

# Studiewijzer

Tristan -> Blauw

Rasmus -> Groen

Max -> Rood

Ferre -> Roze

## 1. Cloud Concepts

### Doelstellingen

De student:

- kan de term "cloud computing" en zijn 3 delen benoemen & in eigen woorden omschrijven.

**Cloud computing** zijn gedeelde pools van **configureerbare computer resources** en high-level **services** die **snel** kunnen **geprovisioned** worden met **minimal management** effort

**SaaS** (Software as a Service)

- Gebruiksklare applicatie in de cloud

**PaaS** (Platform as a Service)

- Gebruiksklaar platform met OS > Vaak voor ontwikkelplatformen

**IaaS** (Infrastructure as a Service)

- Resources > Verantwoordelijk voor eigen OS

- kan de term "server provisioning" benoemen & in eigen woorden omschrijven..

**Server Provisioning** is het opzetten van een server voor gebruik binnen een netwerk, het kan zowel slaan op het installeren van een OS of andere software als op het configureren van de bovenstaande onderdelen.

- kan differentiëren tussen "user perspective" & "back end".

**User perspective:**

- Minimal management
- Minimale tussenkomst van IT staff
- Gemakkelijk provisionen van processen
- Eventueel kost besparend in bepaalde cases

**Back end**

- Complexe networking, storage en server configuraties
- Vaak self-monitoring en self-healing

- kan de verschillen tussen on-premise & cloud benoemen & in eigen woorden omschrijven.

#### **On-premise**

- Lokaal netwerk en servers
- Kostelijk om te onderhouden
- In-house IT support
- Backups eigen verantwoordelijkheid
- Gelimiteerd tot eigen internet connectie
- Scaling en upgradering moeilijk

#### **Cloud Hosted**

- Cloud hosted servers
- Cost per use
- Outsourced IT support
- Automatische backups in meerdere datacenters
- Fiber internet connectie, extreem snel
- Scaling en upgradering makkelijk

- kan op z'n minst 4 grote cloud service providers benoemen.

- AWS
- Azure
- Google Cloud
- VMware Cloud on AWS
- IBM
- Oracle
- Alibaba

- kan op z'n minst 5 (potentiële) voordelen van het gebruik van cloud benoemen & in eigen woorden omschrijven.

- **On-demand self-service**
  - Toegankelijk overal en altijd -> **High Availability**
- **Broad network access**
  - Toegankelijk van elk apparaat over het internet
- **Resource pooling**
  - Provider verantwoordelijk voor hardware
- **Snelle schaalbaarheid**
  - Ook automatisch in bepaalde cases
- **Measured service**
  - Cost per use

- kan 4 nadelen van het gebruik van de cloud benoemen & in eigen woorden omschrijven.

- Afhankelijk van internetverbinding
- Afhankelijk van provider
- Kosten kunnen snel oplopen
- Vendor lock-in

- kan de 4 essentiële karakteristieken van Cloud computing benoemen & in eigen woorden omschrijven.
  - **On-demand self-service**
    - Toegankelijk overal en altijd -> **High Availability**
  - **Broad network access**
    - Toegankelijk van elk apparaat over het internet
  - **Resource pooling**
    - Provider verantwoordelijk voor hardware
  - **Snelle schaalbaarheid**
    - Ook automatisch in bepaalde cases
- kan het verschil tussen “elasticity” & “scalability” benoemen & in eigen woorden omschrijven..

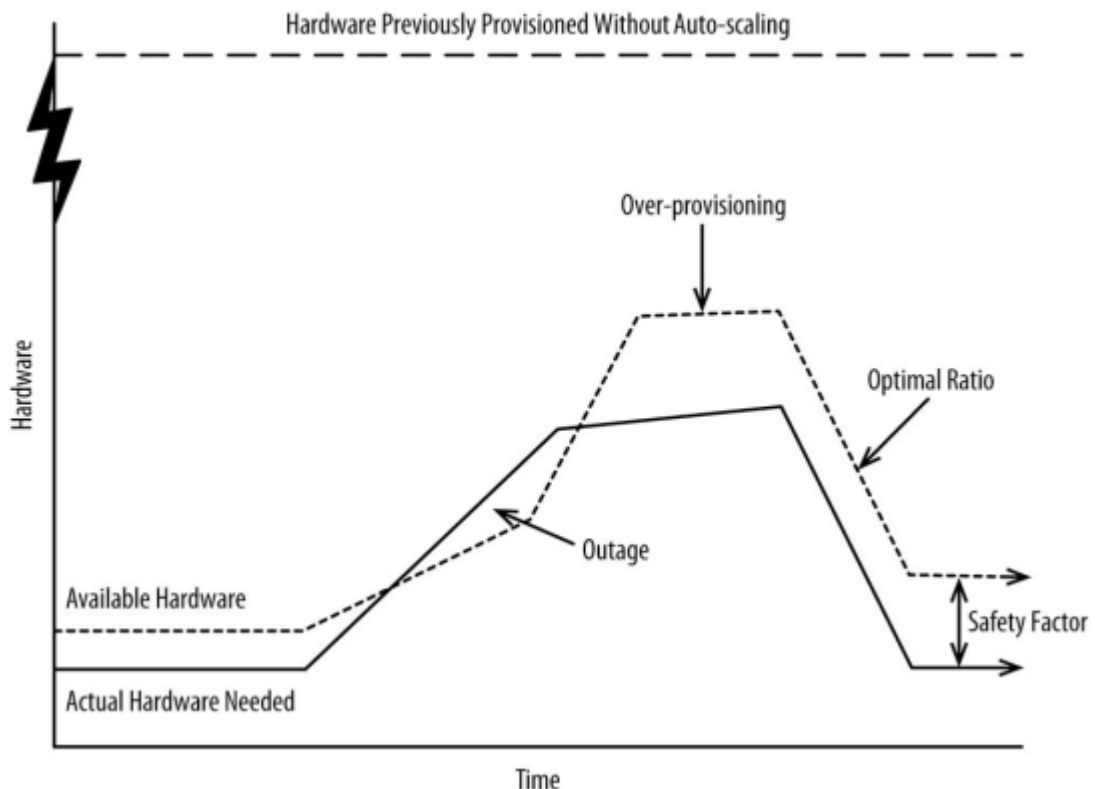
### Elasticity

- De hypervisor aanzetten tot het maken van virtual machines of containers met de benodigde resources om aan de real-time demand te voldoen.

### Scalability

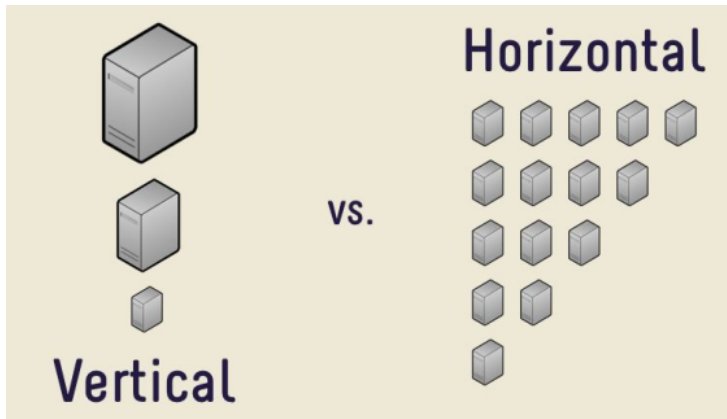
- Vaak op de applicatielaag, een systeem, netwerk of proces dat een groeiend aantal werk kan opvangen en zich potentieel zal uitbreiden

- kan het voordeel van (auto)scaling uitleggen aan de hand van een grafiek.



Wanneer het werk stijgt zal het systeem dit oppikken en voorspellen hoeveel het nodig heeft / gaat hebben. Deze resources zullen ook afnemen wanneer het werk weer daalt.

- kan het verschil tussen “vertical scaling” & “horizontal scaling” benoemen & in eigen woorden omschrijven.



### Vertical scaling

- Het vergroten van een machine, de resources van een machine vergroten.

### Horizontal scaling

- Het vergroten van het aantal machines/containers

- kan de verschillen tussen pets & cattle in een cloud setting benoemen & kan deze linken aan een praktische case.

### Pets

- Hebben een naam
- Opgevoed en uniek
- Wanneer er iets mee scheelt, verzorg je deze tot het weer goed is.

### Cattle

- Hebben een nummer
- Bijna identiek aan elkaar
- Wanneer er iets mee scheelt, neem je een nieuwe en doe je de andere weg.
- In **conclusie**, servers zouden cattle moeten zijn. Zo kan je makkelijk “afscheid” nemen en een nieuwe aanmaken.

- kan de concepten van pets & cattle linken aan verschillende scaling methods.

### Horizontal scaling > Cattle

- Focus op het automatisch toevoegen en verwijderen van server-instanties

### Vertical scaling > Eventueel pets

- Richt zich op het optimaliseren en upgraden van individuele servers

- kan de verschillende cloud service models benoemen & in eigen woorden omschrijven en kan dit linken aan praktische voorbeelden.

**SaaS** (Software as a Service)

- Gebruiksklare applicatie in de cloud

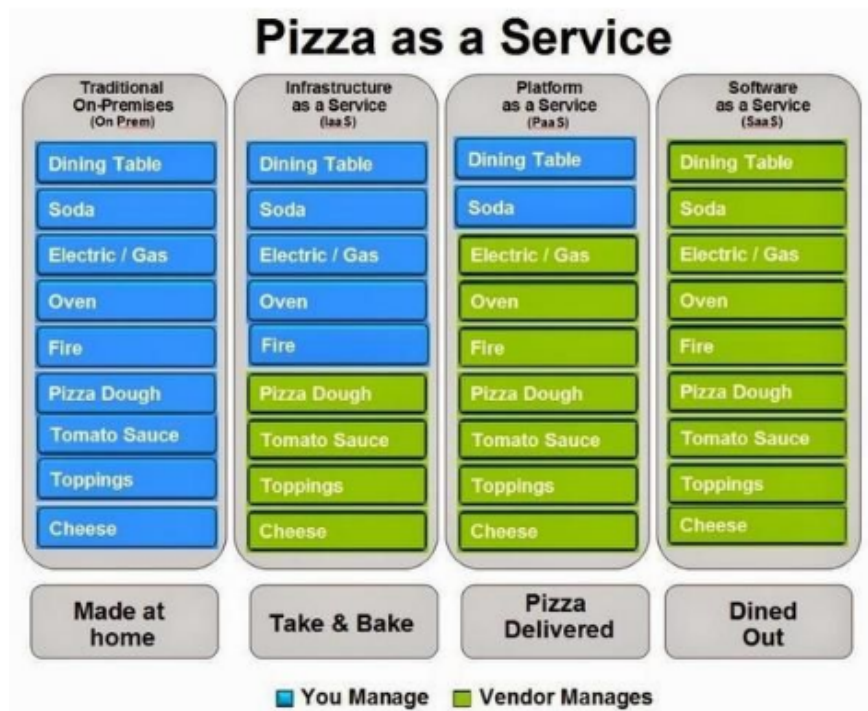
**PaaS** (Platform as a Service)

- Gebruiksklaar platform met OS > Vaak voor ontwikkelplatformen

**IaaS** (Infrastructure as a Service)

- Resources > Verantwoordelijk voor eigen OS

- kan de verschillende personen en hun rol in IT projecten linken aan het correcte service model.



- kan de verschillende "cloud deployment modellen benoemen & in eigen woorden omschrijven.

**Public cloud**

- Cloudomgevingen beheerd door externe serviceproviders, waarbij meerdere gebruikers dezelfde gedeelde infrastructuur en resources delen.

**Private cloud**

- Cloudomgevingen die exclusief eigendom zijn van een enkele organisatie, waarbij de infrastructuur en resources intern worden beheerd en niet worden gedeeld met andere organisaties.

**Hybrid cloud**

- Een combinatie van zowel public als private clouds, waarbij organisaties gebruikmaken van zowel interne resources als externe cloudservices. Hierdoor kunnen ze profiteren van de flexibiliteit en schaalbaarheid van de public cloud, terwijl ze tegelijkertijd controle en beveiliging behouden over gevoelige gegevens in de private cloud.

## 2. Introductie tot AWS

### Doelstellingen

De student:

- kan in eigen woorden de 4 lagen van AWS diensten uitleggen en toelichten aan de hand van een voorbeeld.

#### **Infrastructuurlaag (Infrastructure Layer)**

- Deze laag omvat de fysieke componenten die nodig zijn om cloudservices uit te voeren. Het biedt de basisinfrastructuur waarop andere lagen zijn gebouwd. Voorbeelden van services in deze laag zijn Amazon EC2 (Elastic Compute Cloud) en Amazon S3 (Simple Storage Service).

#### **Platformlaag (Platform Layer)**

- Deze laag biedt tools en diensten die bovenop de infrastructuurlaag worden gebouwd en ontwikkelaars helpen bij het bouwen, implementeren en beheren van applicaties. Voorbeelden van services in deze laag zijn Amazon RDS (Relational Database Service) en AWS Elastic Beanstalk.

#### **Service-laag (Service Layer)**

- Deze laag bevat vooraf gebouwde services die ontwikkelaars kunnen gebruiken om specifieke taken uit te voeren zonder zich zorgen te hoeven maken over de onderliggende infrastructuur. Voorbeelden van services in deze laag zijn Amazon SNS (Simple Notification Service) en AWS Lambda.

#### **Toepassingslaag (Application Layer)**

- Deze laag vertegenwoordigt de applicaties die worden gebouwd en uitgevoerd bovenop de andere lagen. Het omvat aangepaste toepassingen die zijn ontwikkeld met behulp van de beschikbare services en tools van AWS. Voorbeelden van toepassingen in deze laag zijn een e-commercewebsite die gebruikmaakt van EC2, S3 en RDS om producten op te slaan, weer te geven en te verkopen, of een realtime analysesysteem dat gebruikmaakt van SNS en Lambda om gegevens te verwerken en te analyseren.

- kan de 4 foundation services benoemen & in eigen woorden omschrijven.

### Compute:

AWS biedt diensten aan waarmee programma's kunnen worden uitgevoerd in de cloud. Dit kan ook serverless gebeuren door gebruik van lambda's om code uit te voeren.



### Storage:

AWS biedt verschillende vormen van data opslag aan. Uit deze diensten moet gekozen worden afhankelijk van jouw toepassing.



### Network & Content Delivery:

AWS biedt diensten aan om netwerkverkeer te regelen tussen cloud services. Ook biedt het diensten aan voor het assisteren in leveren van content of resources naar de eind-gebruiker.



### Security, Identity & Compliance:

AWS biedt services aan om je gegevens en applicaties te beveiligen, gebruikerstoegang te beheren en te voldoen aan 'industry regulations'.



- kan de 5 Platform services benoemen, in eigen woorden omschrijven en hiervan een voorbeeld geven.

Databases	Analytics	App Services	Deployment & Management	Mobile Services
Relational	Hadoop	Queuing	Containers	Identity
No SQL	Real-time	Orchestration	Dev/ops Tools	Sync
Caching	Data Warehouse	App Streaming	Resource Templates	Mobile Analytics
	Data Workflows	Transcoding	Usage Tracking	Notifications
		Email	Monitoring and Logs	
		Search		

### Databases:



AWS biedt een scala aan databaseservices om te voldoen aan verschillende behoeften op het gebied van gegevensopslag en -beheer.

#### **Analytics:**

Analytics is het proces van het extraheren van inzichten en patronen uit gegevens om waardevolle kennis te verkrijgen en geïnformeerde beslissingen te nemen.

#### **App Services:**

AWS App Services biedt gespecialiseerde functionaliteiten om verschillende aspecten van applicatieontwikkeling en -beheer te ondersteunen. Of het nu gaat om het efficiënt mogelijk maken van 'message queuing', het orchestreren van complexe workflows, het transcoderen van mediabestanden, het beheren van e-mailcommunicatie of het implementeren van krachtige zoekfunctionaliteiten.

#### **Deployment & Management:**

Deployment & Management verwijst naar de reeks services en tools die door AWS worden geleverd om het efficiënt implementeren, schalen, monitoren en beheren van applicaties en resources in de cloud te vergemakkelijken.

#### **Mobile Services:**

Diensten ontworpen om de ontwikkeling en het beheer van mobiele applicaties te ondersteunen.

- *kan de infrastructure laag van AWS diensten linken aan de foundation services.*

#### **Regions:**

Foundational services zoals compute, storage en network worden over meerdere regio's verspreid om 'resillience' te verzekeren en downtime te minimaliseren.

#### **Availability zones:**

Foundational services zoals compute, storage en network worden gerepliceerd over meerdere availability zones binnen elke regio om 'resillience' te verzekeren en downtime te minimaliseren.

## CDN Points Of Pressure:

Foundational services zoals content delivery en network maken gebruik van strategisch geplaatste CDN Points of Presence (PoP's) om efficiënte contentdistributie te verzekeren en latentie voor eindgebruikers te minimaliseren.

- *kan de compute services in eigen woorden omschrijven & kan deze linken aan een praktische case.*
- *kan de storage services in eigen woorden omschrijven & kan deze linken aan een praktische case.*
- *kan het verschil tussen de “snow” storage service en de rest van de storage services benoemen in eigen woorden uitleggen.*

Snow storage maakt gebruik van offline data transfer. Je kan storage devices aanvragen bij aws wanneer de data hierop aanwezig is kan deze opgestuurd worden naar een Amazon data center.

- *kan de Network & Content Delivery services in eigen woorden omschrijven & kan deze linken aan een praktische case.*
- *kan de Security, Identity & Compliance services in eigen woorden omschrijven & kan deze linken aan een praktische case.*
- *kent de voor- en nadelen van het gebruik van “Regions” benoemen & in eigen woorden omschrijven..*

AWS-regio's bieden een breed scala aan voordelen, waaronder hoge beschikbaarheid en mogelijkheden voor disaster recovery. Het is echter van belang om factoren zoals latency, data transfer costs, beschikbaarheid van services in bepaalde regio's en complexiteit van de architectuur in overweging te nemen.

- *kent de voor- en nadelen van het gebruik van “Availability zones” benoemen & in eigen woorden omschrijven.*

Availability zones zorgen voor de mogelijkheid tot high availability, scalability en load balancing. Ook biedt het end users low latency aan dankzij diverse endpoints over de wereld. Rekening moet worden gehouden met de hogere kost, een gelimiteerde hoeveelheid az in elke region en de afhankelijkheid van de region waar de az zich bevindt.

- *kan de concepten van geografische regio's, availability zones en edge network locaties in eigen woorden omschrijven en linken aan een praktische case.*
- *kan de voordelen van een backbone netwerk in functie van de cloud in eigen woorden omschrijven.*

# Why have a backbone network?



## Security

Traffic traverses our infrastructure rather than the Internet



## Availability

Controlling scaling and redundancy



## Reliable performance

Controlling paths customer traffic traverses



## Connecting closer to customers

Avoiding internet "hot spots" or sub-optimal external connectivity

All Region to Region traffic traverses the backbone\*

\* except within the People's Republic of China

Het backbone netwerk biedt connectie tussen de verschillende regio's en az's. Dit maakt het meer secure dan connectie over het internet. Aangezien dit eigen infrastructuur is, kan availability en performance zelf behandeld worden.

- kan de 3 interface methodes om cloud omgevingen te bedienen benoemen en in eigen woorden omschrijven.

AWS Management Console (Website GUI).

AWS Commandline Interface.

AWS Software Developer Kits(SDK's): SDK's stellen ontwikkelaars in staat om AWS-services rechtstreeks in hun applicaties te integreren.

# 3. Compute

## Doelstellingen

De student:

- kan de algemene concepten Hypervisor, Host server, EC2, Instance family, instance size, AMI AWS Graviton, General-purpose instance, Compute-optimized instance, Memory-optimized instance, i/o optimized instance, Burstable instance, Large-scale memory-optimized, EBS, EC2 instance store, security groups, Bastion server ... benoemen en hier voorbeelden van geven.

### **Hypervisor:**

- Softwarelaag die meerdere virtuele machines op een fysieke server mogelijk maakt.

### **Hostserver:**

- Fysieke server waarop virtuele machines worden geïmplementeerd en uitgevoerd.

### **EC2 (Elastic Compute Cloud):**

- AWS-service voor schaalbare virtuele servers in de cloud.

### **Instance family:**

- Groep EC2-instances met vergelijkbare kenmerken en prestaties.

### **Instance size:**

- Specificatie van de rekenkracht, geheugen en opslagcapaciteit van een EC2-instance.

### **AMI (Amazon Machine Image):**

- Een vooraf geconfigureerd sjabloon dat wordt gebruikt om een EC2-instance te starten.

### **AWS Graviton:**

- Een door AWS ontworpen ARM-gebaseerde processorarchitectuur voor EC2-instances.

### **General-purpose instance:**

- EC2-instanceoptie met een evenwichtige mix van rekenkracht, geheugen en netwerkbronnen.

### **Compute-optimized instance:**

- EC2-instanceoptie geoptimaliseerd voor workloads met intensieve rekenprocessen.

**Memory-optimized instance:**

- EC2-instanceoptie met een grote hoeveelheid geheugen voor geheugenintensieve workloads.

**I/O-optimized instance:**

- EC2-instanceoptie geoptimaliseerd voor workloads met intensieve gegevensinvoer en -uitvoer.

**Burstable instance:**

- EC2-instanceoptie met een basale prestatieniveau dat tijdelijk kan worden verhoogd wanneer dat nodig is.

**Large-scale memory-optimized:**

- EC2-instanceoptie met een zeer grote hoeveelheid geheugen voor geheugenintensieve workloads op grote schaal.

**EBS (Elastic Block Store):**

- Een opslagservice voor het aanmaken en beheren van persistente blokopslagvolumes voor EC2-instances.

**EC2 instance store:**

- Tijdelijke opslagruimte op de fysieke server die beschikbaar is voor gebruik door EC2-instances.

**Security groups:**

- Beveiligingsregels die inkomend en uitgaand verkeer naar EC2-instances regelen.

**Bastion server:**

- Een speciale server die wordt gebruikt als beveiligde toegangspoort tot andere servers in een netwerk.

- kan EC2 instance karakteristieken benoemen en uitleggen in eigen woorden.

### **Rekenkracht (Compute Power)**

- EC2-instances hebben verschillende niveaus van rekenkracht, variërend van basisniveaus tot geavanceerde high-performance opties. De rekenkracht wordt meestal gemeten in termen van CPU-kernen en kloksnelheid. Instances met hogere rekenkracht kunnen complexere berekeningen en taken sneller uitvoeren.

### **Geheugen (Memory)**

- De hoeveelheid geheugen die beschikbaar is op een EC2-instance is een belangrijke overweging, vooral voor workloads die grote hoeveelheden gegevens moeten verwerken of in het geheugen moeten houden. Instances met meer geheugen kunnen meer gegevens tegelijkertijd verwerken en bieden betere prestaties voor geheugenintensieve toepassingen.

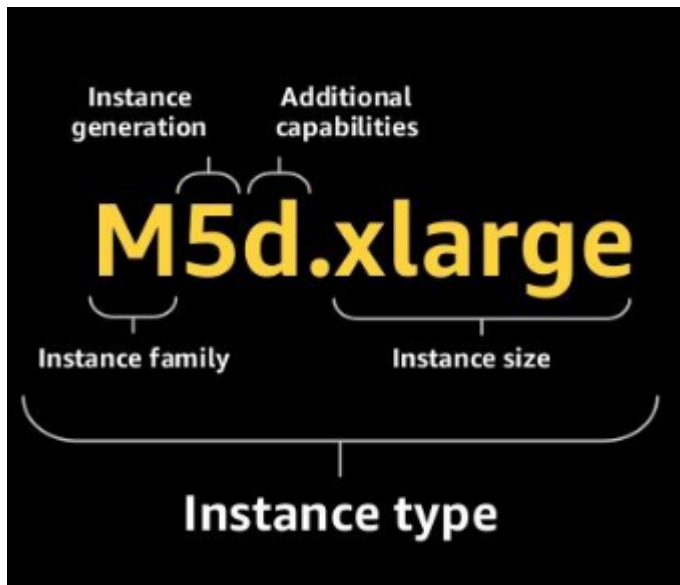
### **Opslag (Storage)**

- EC2-instances bieden verschillende opslagopties, waaronder lokale instantieopslag (instance store) en EBS (Elastic Block Store). Instance store is tijdelijke, lokale opslagruimte die beschikbaar is op de fysieke server van de instance. EBS biedt persistente blokopslag die onafhankelijk is van de levensduur van de instance. De keuze tussen beide hangt af van de vereisten van de toepassing, zoals duurzaamheid, elasticiteit en prestaties.

### **Netwerkprestaties (Network Performance)**

- EC2-instances hebben verschillende niveaus van netwerkprestaties, die de bandbreedte en latentie beïnvloeden. Instances met hogere netwerkprestaties zijn geschikt voor toepassingen met veel netwerkverkeer, zoals web- en databaseservers. Het is belangrijk om het juiste netwerkprestatieniveau te kiezen op basis van de vereisten van de toepassing.

- kan aan de hand van de instance type de instance karakteristieken herleiden en deze linken aan een praktische case



- kan de 3 soorten AMIs benoemen en uitleggen in eigen woorden.

#### **Amazon maintained**

- Up-to-date gehouden door Amazon in elke regio, breed scala aan linux en windows images.

#### **Marketplace maintained**

- Managed en maintained door AWS Marketplace partners

#### **Self maintained**

- Zelfgemaakte AMIs van EC2 instances, kunnen privé gehouden worden, gedeeld met andere of gedeeld worden met de hele community

- kan minstens 3 voordelen van het organiseren van eigen AMIs benoemen & in eigen woorden omschrijven.

#### **Snelle implementatie en consistentie:**

- Geconfigureerde sjablonen maken het mogelijk om snel nieuwe instances te implementeren met de gewenste configuratie en zorgen voor consistentie in het implementatieproces.

#### **Herbruikbaarheid en schaalbaarheid:**

- Georganiseerde AMIs kunnen worden hergebruikt voor meerdere instances en omgevingen, waardoor schaalbaarheid en efficiëntie worden bevorderd.

#### **Aangepaste optimalisatie:**

- Door eigen AMIs te organiseren, kun je optimalisaties en aanpassingen specifiek voor jouw workload toepassen, wat leidt tot betere prestaties en kostenbesparingen.
- kan minstens 3 Categorieën en capabilities van EC2 instances benoemen en uitleggen in eigen woorden.

#### **M5-instances:**

- Algemene computertaken
- Goede balans tussen rekenkracht en geheugen
- Geschikt voor webtoepassingen, app-servers, databases en bedrijfskritische applicaties

#### **C5-instances:**

- Geoptimaliseerd voor compute-intensieve workloads
- Ideaal voor geavanceerde analytics, machine learning en high-performance computing

#### **R5-instances:**

- Geoptimaliseerd voor geheugenintensieve workloads
- Geschikt voor in-memory databases, caching en real-time analyses

#### **I3-instances:**

- Geoptimaliseerd voor storage-intensieve workloads
- Ideaal voor big data, databases en datawarehouses

#### **T3-instances:**

- Burstable performance voor algemene workloads
- Geschikt voor webtoepassingen, ontwikkel- en testomgevingen

#### **X1 & X1e-instances:**

- Geoptimaliseerd voor geheugenintensieve workloads op grote schaal
- Ideaal voor in-memory databases, SAP HANA en real-time analyses op grote datasets



- kan de verschillende inlogmethodes voor EC2 instances benoemen en uitleggen in eigen woorden.

### **SSH (Secure Shell)**

**RDP (Remote Desktop Protocol):** RDP is de standaardmethode om op afstand toegang te krijgen tot Windows-gebaseerde EC2-instances. Met RDP kun je het bureaublad van de instance bekijken en deze op dezelfde manier bedienen als een fysieke computer. Je hebt een RDP-client nodig om verbinding te maken met de instance en je kunt inloggen met een gebruikersnaam en wachtwoord.

**AWS Systems Manager Session Manager:** Session Manager is een service van AWS Systems Manager waarmee je veilig op afstand verbinding kunt maken met EC2-instances. Het maakt gebruik van het Secure Shell Protocol (SSH) of het Secure Sockets Layer (SSL) protocol om versleutelde sessies tot stand te brengen. Hiermee kun je toegang krijgen tot de command-line interface (CLI) van de instance zonder dat je openbare IP-adressen of inkomende regels voor de beveiligingsgroep hoeft te configureren.

**AWS Management Console:** Via de AWS Management Console kun je ook inloggen op je EC2-instances. Je kunt inloggen op de EC2-servicepagina, de gewenste instance selecteren en vervolgens een instantieverbinding starten. Deze methode biedt een grafische gebruikersinterface (GUI) waarmee je de instance kunt beheren zonder een externe terminalclient te gebruiken.

## 4. Storage

### Doelstellingen

De student:

- kan de concepten file storage, Block storage, Object storage, HA File storage ... in eigen woorden uitleggen en kan hiervan concrete voorbeelden geven.
- kan de 3 voordelen van file storage benoemen en uitleggen in eigen woorden.

#### **File Storage**

Data wat opgeslagen is als files in een directory structuur/hiërarchie. Ze zijn shared over een netwerk, toegankelijk via een hiërarchisch pad, zoals mappen en submappen.

Populair vanwege:

- 'works natively with OS'
- 'provides shared access while providing consistency guarantees and locking functionality'
- 'provides hierarchical namespace'

**Voorbeeld:** Een NAS (Network Attached Storage), waarbij meerdere computers toegang hebben tot een gedeeld FS (File System).

#### **Block Storage**

Block storage slaat gegevens op als blokken met een vaste grootte. Deze blokken kunnen worden behandeld als individuele entiteiten en kunnen willekeurig worden gelezen of beschreven.

**Voorbeelden:** Het wordt vaak gebruikt bij het bouwen van harde schijven, SSD's en Storage Area Networks (SAN's). Een ander voorbeeld van block storage is het gebruik van een externe harde schijf die via USB is aangesloten op een computer.

#### **Object Storage**

Object storage slaat gegevens op als objecten, elk met een unieke ID (zoals bijv MongoDB). 'Simple API to get and put data based on key(id)'

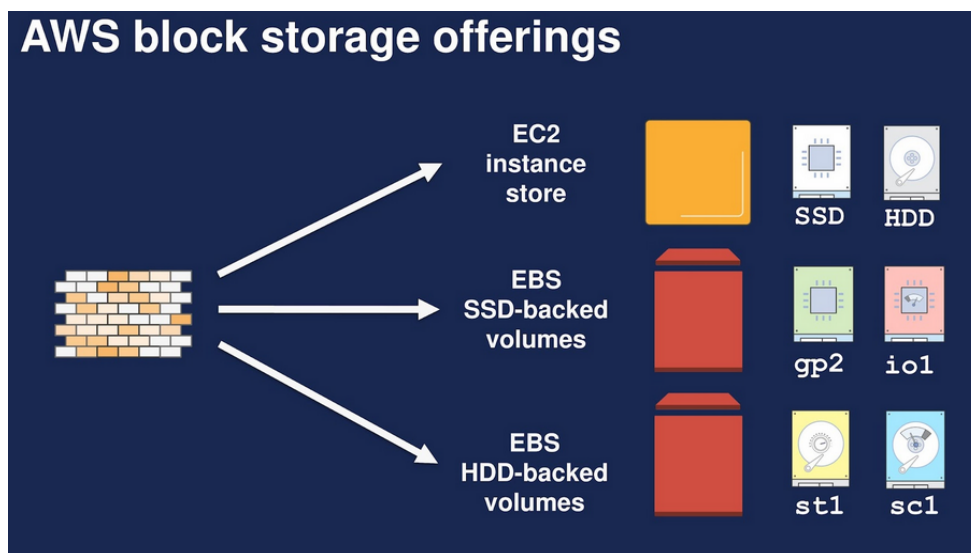
**Voorbeelden:** Amazon S3 (Simple Storage Service), waarin je objecten kunt opslaan en benaderen via een unieke URL. Object storage is ideaal voor het opslaan van grote hoeveelheden ongestructureerde gegevens, zoals afbeeldingen, video's, documenten en back-upbestanden.

## HA File Storage

Highly Available File Storage maakt gebruik van redundancy om voor H.A. te zorgen, door bijvoorbeeld gebruik te maken van replicatie en failover-mechanismen.

**Voorbeelden:** Een voorbeeld van HA file storage is een oplossing waarbij bestanden worden gerepliceerd naar meerdere locaties, bijvoorbeeld tussen twee EC2's. Als ene uitvalt, is de data nog steeds beschikbaar via de andere.

- Kan de verschillende block storage oplossingen binnen AWS benoemen, kan hier voorbeelden van geven en kan een case analyseren om hierin de juiste oplossing toe te passen.



- kan de eigenschappen van EC2 Instance store benoemen en in eigen woorden uitleggen.
- kan de eigenschappen van EBS benoemen en in eigen woorden uitleggen.

### **Opsomming/Voorbeelden:**

EC2 instance store: tijdelijke opslag die fysiek gekoppeld is aan een hostmachine. Voorbeeld: Stel dat je een applicatie hebt die tijdelijke gegevens verwerkt, zoals cachegegevens of tijdelijke berekeningen. Omdat de opslag fysiek is gekoppeld aan de hostmachine, biedt het een zeer lage latency en hoge throughput. Andere eigenschappen: Local to instance, non-persistent data store, data not replicated, no snapshot support, SSD/HDD.

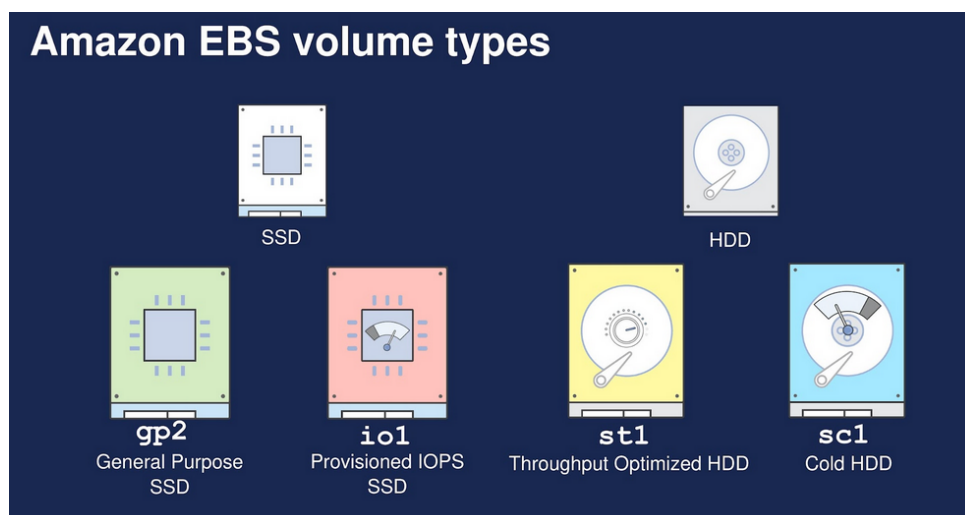
EBS SSD-backed volumes: Elastic Block Store biedt persistent opslag met hoge prestaties. Bijvoorbeeld, als je een online winkel hebt met veel transacties, kan het gebruik van SSD EBS-volumes de responstijden verbeteren en de algehele prestaties van de applicatie verhogen. Andere eigenschappen: Create/Attach volumes through an API, Service access over network.

EBS HDD-backed volumes: goedkoper, maar minder krachtig i.v.m. SSD-backed. Voorbeeld: Stel dat je een back-up opslag of een data lake hebt waarbij kosten een belangrijke factor zijn en hoge prestaties geen absolute vereiste zijn, bijvoorbeeld, als je archieven of log bestanden opslaat waarbij frequente toegang niet nodig is.

- kan minstens 2 voor- en nadelen benoemen van EBS en in eigen woorden uitleggen.

Voordelen	Nadelen
Data Persistence	Cost: can be expensive
High Availability	Performance Limitations: might not meet performance needs of extremely I/O-intensive applications
Flexibility & Scalability	Regional Dependency: tied to specific AWS region and cannot be easily replicated across regions (not the case for availability zones within one region)

- kan onderscheid maken tussen de verschillende EBS volume types en deze in eigen woorden uitleggen en linken aan een praktische case.
- kan de Elastic Volumes eigenschappen benoemen en in eigen woorden uitleggen.



**gp2:** 'General Purpose' SSD EBS-volumes bieden een goede balans tussen prijs en prestatie. Ze leveren basisprestaties en burst-mogelijkheden, waardoor ze geschikt zijn voor een breed scala aan workloads met average I/O-behoeften.

Voorbeelden: Bootvolumes voor EC2-instances, kleine tot middelgrote databases, low-latency interactieve applicaties.

**io1:** Provisioned IOPS SSD EBS-volumes bieden krachtige opslag met aanpasbare IOPS. Ze zijn ontworpen voor I/O-intensieve workloads zoals grote databases en applicaties die consistente en lage latency-prestaties vereisen.

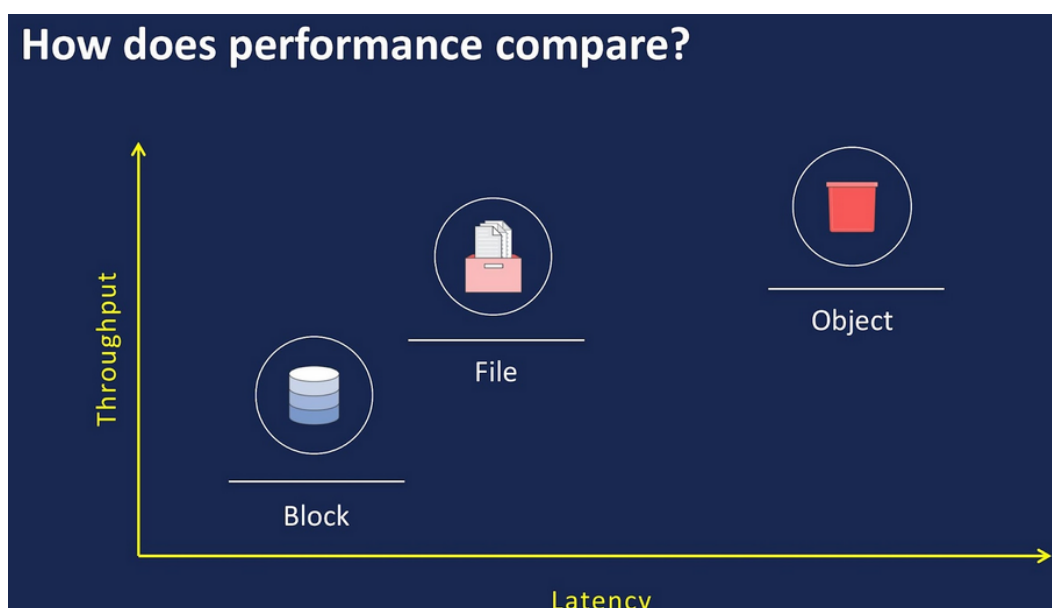
Voorbeelden: Grote databases met hoge transactievolumes, intensieve applicaties voor gegevensverwerking.

**st1:** Throughput Optimized HDD EBS-volumes bieden hoogwaardige en kosteneffectieve opslag voor grote, sequentiële I/O-workloads. Ze zijn zeer geschikt voor big data, datawarehouses, logverwerking en streaming-workloads.

**sc1:** Cold HDD EBS-volumes zijn ontworpen voor sporadisch benaderde, throughput-georiënteerde workloads. Ze bieden goedkope opslag met lagere IOPS en langere hersteltijden, waardoor ze geschikt zijn voor grote, sequentiële lees-/schrijfoperaties, back-ups en archiefopslag.

Voorbeelden: Back-ups, archiefopslag, gegevensopslag voor zelden gebruikte gegevens.

- kan de performantie van de verschillende storage types in eigen woorden uitleggen aan de hand van een grafiek.



Een hogere throughput duidt op het vermogen om een grotere hoeveelheid gegevens per tijdseenheid over te dragen, wat wijst op snellere gegevensverwerking of -overdracht. Een lagere latentie (latency) impliceert verminderde vertraging of wachttijd bij het benaderen van gegevens of het ontvangen van een reactie, wat resulteert in snellere en responsieve operaties.

- kan de voordelen van EFS benoemen aan de hand van een vergelijking met een DIY file storage oplossing.

### ***Voordelen:***

1. EFS biedt een eenvoudige en schaalbare opslaginfrastructuur zonder dat er zorgen zijn over hardware configuratie en -onderhoud. DIY FileStorage kan dat niet.
  2. Het biedt ook automatische schaling, waardoor je opslagcapaciteit naadloos kan worden vergroot of verkleind op basis van de behoeften van je applicatie. DIY FileStorage kan dat niet.
  3. EFS zorgt ook voor High Availability en duurzaamheid door gegevens over meerdere zones te repliceren.
  4. Daarnaast biedt EFS geavanceerde beveiligingsfuncties, zoals toegangsbeheer en encryptie, om de gegevens veilig te houden. Dit moet handmatig geconfigureerd worden bij een DIY oplossing.
- kan de 3 basis eigenschappen van S3 storage benoemen, in eigen woorden uitleggen en linken aan een praktische case.

## **S3 Buckets**

### ***Eigenschappen:***

- Collect: Het kan gegevens ontvangen van verschillende bronnen, zoals webapplicaties, IoT-apparaten, logbestanden of pipelines. S3 biedt flexibele API's en SDK's waarmee ontwikkelaars gegevens rechtstreeks naar S3-buckets kunnen uploaden.
- Store: S3 biedt High Availability, wat betekent dat gegevens redundant worden opgeslagen op meerdere apparaten binnen een AWS-regio.

- **Analyze:** S3 integreert naadloos met verschillende AWS-analytische services. Hierdoor kunnen gebruikers complexe gegevensanalyse, query's en verwerking rechtstreeks uitvoeren op de gegevens die zijn opgeslagen in S3

***Built for:***

- Backup & Restore
  - Data lakes & Analytics
  - Cloud-native applications
- kan de 3 basis eigenschappen van Glacier storage benoemen, in eigen woorden uitleggen en linken aan een praktische case.

**Amazon Glacier**

***Eigenschappen:***

- **Secure:** maakt gebruik van encryption tijdens overdracht en opslag om de vertrouwelijkheid en integriteit van de gegevens te waarborgen.
- **99.999999999% Durable:** Amazon Glacier maakt gebruik van redundante opslag binnen meerdere AWS-regio's om gegevensverlies te voorkomen.
- **Cost-effective:** Het biedt lagere opslagkosten in vergelijking met andere AWS-opslagdiensten, zoals Amazon S3, en biedt verschillende storage-tiers met variabele prijsniveaus, afhankelijk van de vereiste hersteltijd.

***Built for:***

- Active archive
  - Tape replacement
  - Regulatory Compliance
- kan de storage lifecycle in eigen woorden uitleggen en de verschillende stappen benoemen en linken aan een praktische case.

“An S3 Lifecycle configuration is an XML file that consists of a set of rules with predefined actions that you want Amazon S3 to perform on objects during their lifetime. Amazon S3 provides a set of REST API operations for managing lifecycle configuration on a bucket. Amazon S3 stores the configuration as a *lifecycle subresource* that is attached to your bucket. “

**Type of actions:**

1. Transition actions
2. Expiration actions

**Praktische case:**

Stel dat een organisatie regelmatig logbestanden uploadt naar een S3-bucket voor het bijhouden van applicatieactiviteit. Deze logbestanden zijn gedurende een week nodig voor analyse en probleemoplossing. In dit geval kan de organisatie een S3 Lifecycle-configuratie instellen. De configuratie bevat regels die Amazon S3 instrueren om de logbestanden automatisch te verplaatsen naar een minder dure opslagklasse, zoals S3 Standard-IA (Infrequent Access), na een week

- kan minstens 2 voor- en nadelen van Glacier benoemen en in eigen woorden uitleggen.

Voordelen	Nadelen
Kosten: lage opslagkosten, goede cost-effective keuze voor lange termijn	De retrieval van gegevens duurt vrij lang, van enkele minuten tot uren zelfs
Highly Available	Opslagbeleid: Specifieke regels voor gegevensopslag vanwege de regulatory compliance, daardoor eist het wat planning om mee te starten.
Compliance-ondersteuning, dit maakt het geschikt voor het opslaan van gevoelige gegevens met strenge compliance requirements	Amazon Glacier biedt geen directe toegang tot gegevens zoals S3. Het vereist een herstelproces waarbij gegevens naar een S3-bucket moeten worden hersteld voordat ze kunnen worden opgehaald.

- kan minstens 2 voor- en nadelen van S3 benoemen en in eigen woorden uitleggen.

Voordelen	Nadelen
Geautomatiseerde, snelle en betrouwbare asynchrone replicatie van gegevens in AWS-regio's	Kosten kunnen toenemen naarmate de gegevensgrootte en het gebruik.
Schaalbaarheid naar behoefte	S3 bucket policies zijn minder user-friendly
High Availability	Latency: Vanwege de distributed storage kan enige latency optreden bij het opvragen daarvan



- kan het gebruik van S3 & Glacier vertalen naar een praktische case.
- 
1. S3: Het hosten van een website, denk aan PE
  2. Glacier: Archivering van financiële gegevens voor een bedrijf. Amazon Glacier biedt een kosteneffectieve oplossing voor het op lange termijn opslaan van financiële documenten, oude back-ups en compliance-gegevens. Hierdoor kan het bedrijf voldoen aan de wettelijke storage regulations en tegelijkertijd kosten besparen op opslag.

## 5. Networking (les 5+6)

### Doelstellingen

De student:

- kan de concepten VPC, High Availability, public/private subnetting Route 53, RFC1918, CIDR, Dual-stack, Route table, Internet Gateway, Route 53 resolver, Private hosted dns zones, NACL, Flow Logs, VPC Peering, NAT Gateway, Transit gateway ... in eigen woorden uitleggen en kan hiervan concrete voorbeelden geven.

**VPC:** Virtual Private Cloud is een virtueel netwerk in de cloud. Bijvoorbeeld, een VPC kan worden gebruikt om een geïsoleerde omgeving te creëren waarin applicaties kunnen draaien.

**High Availability:** Het vermogen van een systeem om te blijven functioneren, zelfs als er componenten of services uitvallen.

**Public/Private subnetting Route53:** Het onderverdelen van een VPC in publieke en private subnets met behulp van Route53 DNS-service voor het beheren van domeinnamen en het routeren van verkeer. Bijvoorbeeld, het configureren van een publiek subnet voor webtoepassingen en een privé subnet voor database-servers.

**RFC1918:** Een reeks IP-adresbereiken (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) die zijn gereserveerd voor privénetwerken. Bijvoorbeeld, het gebruik van RFC1918 IP-adressen in een VPC voor interne communicatie tussen bronnen.

**CIDR:** Classless Inter-Domain Routing. Bijvoorbeeld, 192.168.0.0/24 is een CIDR-notatie voor een subnet met 256 IP-adressen.

**Dual-stack:** Het ondersteunen van zowel IPv4 als IPv6 in een netwerkomgeving.

**Route table:** Een verzameling regels die bepalen hoe netwerkverkeer wordt doorgestuurd binnen een VPC.

**Internet Gateway:** Een service die het verkeer tussen een VPC en het internet mogelijk maakt.

**Route 53 resolver:** Een functie van Amazon Route 53 waarmee DNS-verzoeken kunnen worden opgelost tussen on-premises netwerken en VPC's.

**Private hosted DNS-zones:** Een functie van Amazon Route 53 waarmee het hosten van privé-DNS-zones binnen een VPC mogelijk is.

**NACL:** Network Access Control List is een firewallregelset op subnetniveau die inkomend en uitgaand verkeer kan filteren.

**Flow Logs:** Logbestanden die informatie bevatten over het inkomende en uitgaande verkeer binnen een VPC.

**VPC Peering:** Het koppelen van twee VPC's zodat ze rechtstreeks met elkaar kunnen communiceren via het Amazon-netwerk. Bijvoorbeeld, het opzetten van VPC-peering tussen twee VPC's in verschillende regio's.

**NAT Gateway:** Network Address Translation Gateway is een beheerde service waarmee instances in een privésubnet toegang kunnen krijgen tot internetbronnen. Bijvoorbeeld, het gebruik van een NAT Gateway om EC2-instances in een privésubnet te laten communiceren met het internet voor software-updates.

**Transit Gateway:** Een gedistribueerd netwerkcomponent dat connectiviteit biedt tussen VPC's en on-premises netwerken.

- kan een H.A. VPC opbouwen en de genomen beslissingen toelichten. → PE?
  - kan de componenten van een VPC benoemen en in eigen woorden uitleggen.
- 
- IPv4/IPv6 Address Blocks
  - Subnet Creation
  - Route tables
  - Internet Connectivity
  - Elastic IP Addresses
  - Network/Subnet security
  - VPN's
  - VPC peering
- 
- kan toelichten wat de criteria zijn waarop gelet moet worden bij het toekennen van een ip range en subnets binnen een VPC

**Omvang:** De range moet groot genoeg zijn voor alle bronnen binnen de VPC, inclusief toekomstige groei en schaalbaarheid. Probeer ook ranges niet te kiezen die voor overlap zorgen met andere netwerken waarmee verbindingen worden gemaakt.

**Scheiding:** Plan de subnetstructuur zodanig dat deze schaalbaar is. Denk na over toekomstige behoeften en eventuele uitbreiding van het netwerk, zodat subnetten gemakkelijk kunnen worden aangepast of toegevoegd zonder onderbreking van de bestaande resources.

**Security:** Gebruik subnetten in combinatie met Network Access Control Lists (NACLs) en beveiligingsgroepen om een veiligheidsperimeter te creëren en inkomend/uitgaand verkeer te beheren. Alternatief: segmenteer resources a.d.h.v. public/private subnets, voorbeeld: frontend resources in de public SN's, en vice versa.

- kan aangeven op welke manier we gebruik kunnen maken van Availability Zones binnen een VPC en in eigen woorden minstens 2 voor- en nadelen uitleggen.

Voordelen	Nadelen
High Availability	Kosten: Bij het repliceren en redundant maken van resources moeten extra middelen worden opgezet
Scalability: Door resources over AZ's te verdelen, wordt het gemakkelijker om de infrastructuur horizontaal op te schalen.	Complexiteit: Het beheren van resources over meerdere AZ's kan complex worden uiteindelijk

- kan aangeven op welke manier we gebruik kunnen maken van Route 53 binnen een VPC en hybrid cloud opstelling.

Route 53 kan worden gebruikt als een private DNS-service binnen de VPC. Het kan interne domeinnaamresolutie bieden voor resources binnen de VPC, zoals EC2-instanties en load balancers. In een hybride cloud-opstelling kan Route 53 worden gebruikt als een DNS-oplossing om zowel de VPC-omgeving als externe resources te beheren.

- kan de verschillen tussen Security Groups & Network ACLs toelichten en in eigen woorden uitleggen.

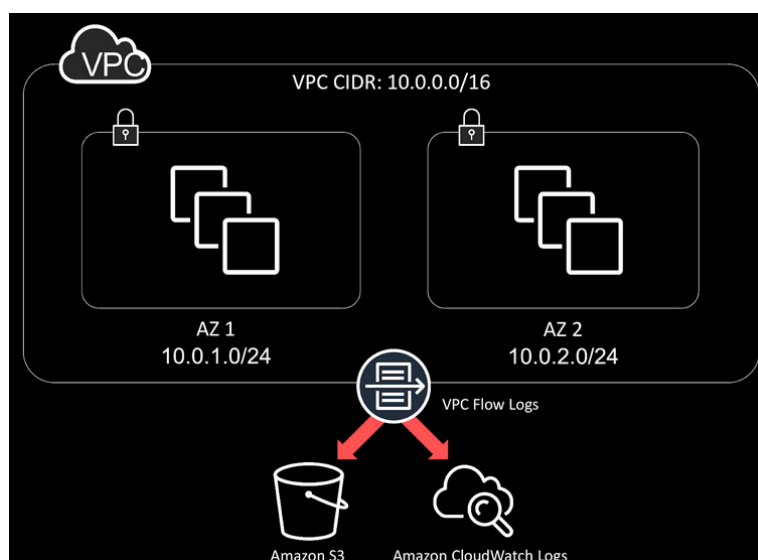
In essentie komt het neer op het volgende: Security Groups zijn gericht op het beveiligen van AWS-bronnen door inkomend en uitgaand verkeer te regelen op basis van de bronnen, terwijl Network ACLs subnetniveau beveiliging bieden door regels toe te passen op het verkeer dat in en uit het subnet gaat.

- kan aangeven op welke manier je services kan linken aan security groups en deze op een veilige manier kan openstellen naar het internet.
- kan in eigen woorden uitleggen op welke manieren security groups gelinkt kunnen worden aan services binnen AWS.

Binnen de AWS Console is het mogelijk om alleen bepaalde andere Security Groups of instances/services binnen te laten. Op deze manier is het mogelijk om alleen essentiële endpoints bloot te stellen aan het internet, zoals de HTTP:80 endpoint van een bepaalde load balancer die de front-end toont.

- kan in eigen woorden de werking van VPC flow logs toelichten en deze linken aan een use case.

VPC Flow Logs genereert gedetailleerde logboeken van inkomend en uitgaand verkeer, inclusief IP-adressen, poorten en protocollen. Deze logboeken bieden inzicht in het netwerkverkeer en kunnen worden gebruikt voor beveiligingsanalyse, probleemoplossing en auditing. Een use case voor VPC Flow Logs is het detecteren van verdacht verkeer of ongeautoriseerde toegangspogingen door de logboeken te analyseren en afwijkingen te identificeren, waardoor je proactieve maatregelen kunt nemen om de beveiliging van je VPC te versterken.



- kan aangeven op welke manier we gebruik kunnen maken van connectivity gateways om op verschillende manieren intern toegang te geven tot een VPC.

Connectivity Gateways zoals Virtual Private Gateway (VGW), Direct Connect Gateway (DXGW) en Transit Gateway (TGW) bieden verschillende methoden voor interne toegang tot een VPC:

- VGW: Versleutelde VPN-verbinding via het openbare internet voor veilige toegang tot resources in de VPC.
  - DXGW: Privéverbinding via een directe fysieke verbinding voor snelle, betrouwbare en veilige connectiviteit.
  - TGW: Geaggregeerd netwerkpunt dat connectiviteit tussen VPC's en on-premises netwerken mogelijk maakt, waardoor hub-and-spoke-architectuur ontstaat.
- kan toelichten hoe we VPC Peering gebruiken in een AWS multi-cloud omgeving en de voordelen hiervan uitleggen aan de hand van een use case.

VPC Peering stelt klanten in staat om veilige netwerkverbindingen tot stand te brengen tussen VPC's in verschillende AWS-accounts of regio's. Op deze manier kunnen resources opgesplitst worden a.d.h.v. VPC's. Het is niet mogelijk om tussen cloud providers VPC peering tot stand te brengen.

- kan de verschillen tussen VPC Peering en een Transit gateway in eigen woorden uitleggen en toelichten met een voorbeeld.

VPC Peering is een directe verbinding tussen twee VPC's voor het routeren van verkeer binnen AWS. Transit Gateway is een gecentraliseerde hub waarmee meerdere VPC's en externe netwerken kunnen worden verbonden. VPC Peering is geschikt voor directe communicatie tussen specifieke VPC's met hoge bandbreedte. Transit Gateway is handig voor complexe netwerkarchitecturen met meerdere VPC's en biedt schaalbaarheid en eenvoudig beheer. Bijvoorbeeld, een organisatie kan Transit Gateway gebruiken om verkeer tussen verschillende VPC's in verschillende regio's te consolideren en te beheren.

## 6. Services - Databanken

### Doelstellingen

De student:

- kan de concepten scale OUT, scale UP, RDS, Aurora, DynamoDB, Elasticsearch, OpenSearch ... in eigen woorden uitleggen en kan hiervan concrete voorbeelden geven.

**Scale OUT:** Het verhogen van de capaciteit door meer identieke bronnen toe te voegen, zoals het toevoegen van extra servers om de belasting te verdelen.

**Scale UP:** Het vergroten van de capaciteit van een enkele bron, zoals het upgraden van een server met krachtigere hardware.

**RDS** (Relational Database Service): Een beheerde database service die het gemakkelijk maakt om relationele databases, zoals MySQL of PostgreSQL, op te zetten, schalen en beheren.

**Aurora:** Een schaalbare en zeer beschikbare relationele database-engine die compatibel is met MySQL en PostgreSQL.

**DynamoDB:** Een volledig beheerde NoSQL-database service die schaalbaarheid, hoge beschikbaarheid en snelle prestaties biedt.

**Elasticsearch:** Een gedistribueerd zoek- en analytisch engine dat geavanceerde zoekmogelijkheden en schaalbaarheid biedt voor het doorzoeken van grote hoeveelheden ongestructureerde gegevens.

**OpenSearch:** Een op Elasticsearch gebaseerde zoek- en analytisch engine die open source en gemeenschapsgericht is, en wordt gebruikt voor het indexeren en doorzoeken van gegevens.

- kan in eigen woorden de verschillende databank types uitleggen en linken aan praktische use cases.

**Amazon RDS:** Een beheerde relationele database service voor het hosten van databases zoals MySQL, PostgreSQL en Oracle.

**Amazon DynamoDB:** Een volledig beheerde NoSQL-database service die schaalbaarheid en hoge beschikbaarheid biedt.

**Amazon DocumentDB:** Een volledig beheerde documentdatabase compatibel met bijv. MongoDB.

**Amazon Neptune:** Een volledig beheerde graphdatabase service voor het opslaan en queryen van zeer verbonden gegevens.

**Amazon Elasticsearch Service:** Een beheerde zoek- en analytische engine voor het doorzoeken en analyseren van ongestructureerde gegevens.

**Amazon Timestream:** Een beheerde tijdreeksdatabase voor het opslaan en analyseren van tijdgebaseerde gegevens.

**Amazon Quantum Ledger Database:** Een volledig beheerde, onveranderlijke en cryptografisch verifieerbare transactiedatabase.

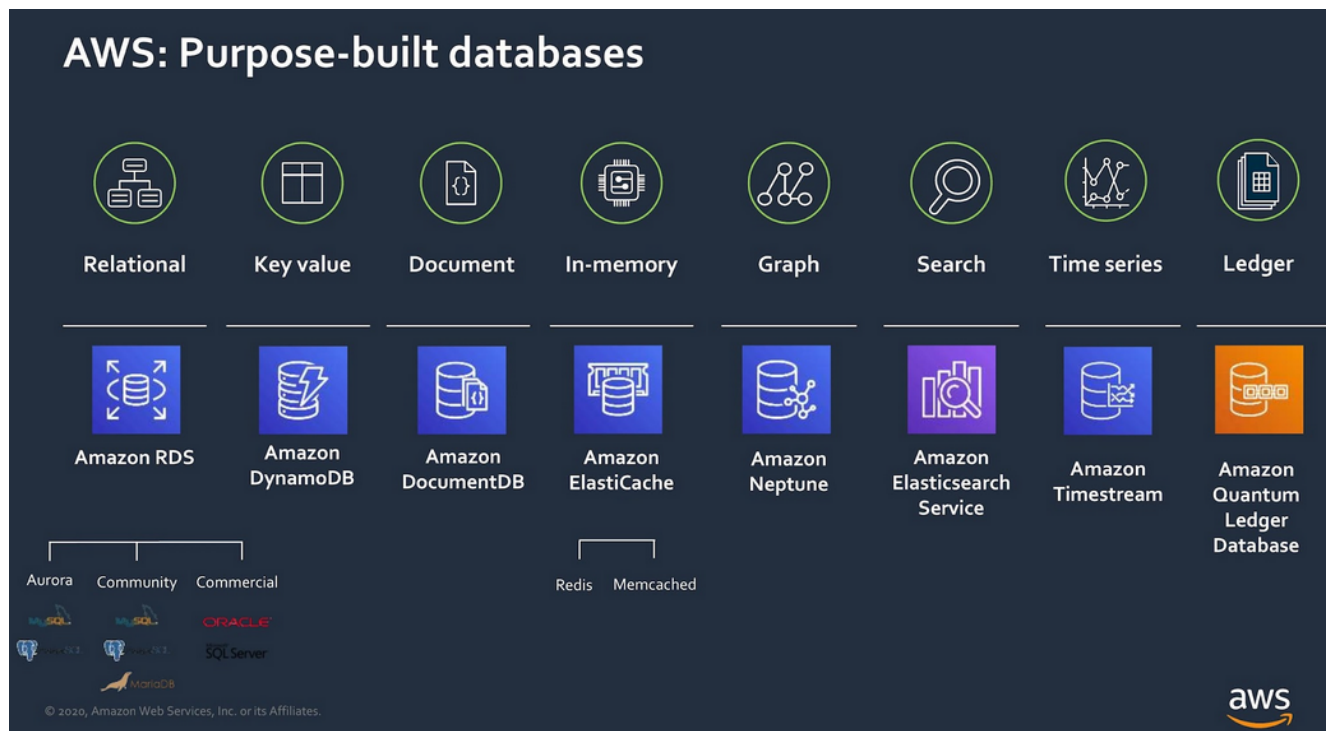
- kan het ACID principe van databanken in eigen woorden uitleggen en de 4 elementen benoemen.

### **Atomic, Consistent, Isolated, Durable**

Atomic zorgt ervoor dat een transactie ofwel volledig wordt uitgevoerd, of helemaal niet. Consistency zorgt ervoor dat een transactie de database van de ene geldige staat naar een andere geldige staat brengt. Isolation zorgt ervoor dat transacties onafhankelijk van elkaar worden uitgevoerd en geen interferentie hebben. Durability zorgt ervoor dat eenmaal bevestigde transacties permanent en persistent zijn, zelfs bij stroomuitval of herstart van de database.



- kan de verschillende databank types linken aan specifieke AWS diensten.



- kan aangeven waarom we sharding bij databanken gebruiken en hoe dit conceptueel in zijn werk gaat.

Sharding houdt in dat de gegevens worden verdeeld over meerdere databaseservers, elk met een subset van de gegevens. Dit verhoogt de parallele verwerkingscapaciteit en vermindert de belasting op individuele servers.

- kan omschrijven welke 4 voordelen RDS diensten hebben over eigen geïnstalleerde databanken.

**Schaalbaarheid:** RDS-diensten bieden eenvoudige schaalbaarheidsopties, zoals het automatisch toevoegen van rekenkracht en opslagruimte.

**Security:** RDS-diensten implementeren beveiligingsmaatregelen, zoals encryption, toegangscontrole en automatische patches.

**High Availability:** RDS-diensten bieden ingebouwde replicatie en failover-mogelijkheden.

**Beheer en Onderhoud:** RDS-diensten zorgen voor het beheer en onderhoud van de database-infrastructuur.

- kan de verschillen tussen Aurora en MySQL RDS instanties benoemen.

Aspect	Aurora	MySQL RDS
Performance	Offers better performance with optimized storage architecture	Standard performance capabilities of MySQL
Scalability	Highly scalable, automatically adjusts to application needs	Can be manually scaled based on resource requirements
Replication	Efficient replication method with fast failover and recovery	Replication capabilities for high availability and disaster recovery
Cost-effectiveness	Can be cost-effective due to optimized resource utilization	Standard pricing model for MySQL RDS instances
Compatibility	Compatible with MySQL, allows easy migration	Fully compatible with MySQL applications


- kan de belangrijkste 3 verschillen tussen relationele en key databases in eigen woorden benoemen.
1. **Structuur en dataorganisatie:** Relationele databases maken gebruik van een gestructureerd schema waarin gegevens worden georganiseerd in tabellen met vooraf gedefinieerde kolommen en relaties tussen tabellen. Key-value databases daarentegen slaan gegevens op als een verzameling van key-values.
  2. **Query-taal en complexiteit:** Relationele databases gebruiken SQL. Key-value databases daarentegen bieden meestal eenvoudigere query-operaties, zoals het ophalen van gegevens op basis van een specifieke key.
  3. **Schaalbaarheid en prestaties:** Relationele databases zijn over het algemeen goed geschikt voor complexe transactiegerichte toepassingen, maar kunnen moeite hebben met schaalbaarheid bij grote datasets en hoge belasting. Key-value databases daarentegen zijn ontworpen voor snelle gegevensopslag en -opvraging

- kan minstens 1 key feature van DynamoDB dat het onderscheid van RDS in eigen woorden benoemen en toelichten.

## Amazon DynamoDB


### Fast and flexible key value database service for any scale

Performance at scale




Consistent, single-digit millisecond response times at any scale; build applications with virtually unlimited throughput

Serverless




No server provisioning, software patching, or upgrades; scales up or down automatically; continuously backs up your data

Comprehensive security




Encrypts all data by default and fully integrates with AWS Identity and Access Management for robust security

Global database for global users and apps



Build global applications with fast access to local data by easily replicating tables across multiple AWS Regions

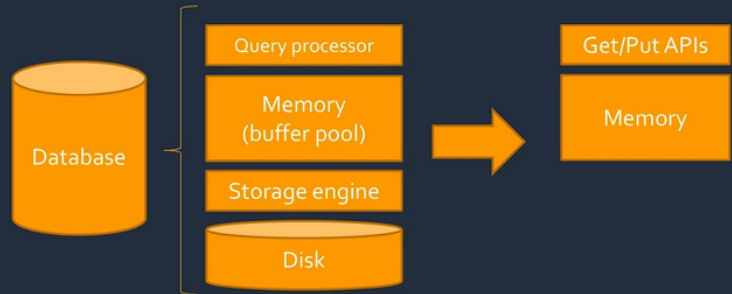
© 2020, Amazon Web Services, Inc. or its Affiliates.



- kan in eigen woorden de eigenschappen van een in-memory databank benoemen.

## In-memory

- No persistence, in-memory
- Microsecond performance
- Simple commands for manipulating in memory data structures
  - Strings, hashes, lists, sets, and sorted sets




```

graph LR
    DB[(Database)] --> QP[Query processor]
    QP --> MP[Memory (buffer pool)]
    MP --> SE[Storage engine]
    SE --> D[(Disk)]
    QP --> G[Get/Put APIs]
    MP --> M[Memory]
  
```

Milliseconds to microseconds (10x faster)

© 2020, Amazon Web Services, Inc. or its Affiliates.



- kan de ElastiCache service in eigen woorden uitleggen.

“Fully compatible with open source Redis & Memcached, In-memory data store and cache for microsecond response times, Network Isolation, encryption at rest or in transit, easily scalable, writes and reads with sharding and replicas”

## 7. Services - Lambda

### Doelstellingen

De student:

- kan de concepten serverless, containers, lambda deployment package, concurrent executions, lambda image resizer, API key, ... in eigen woorden uitleggen en kan hiervan concrete voorbeelden geven.

#### **Serverless:**

- Je hoeft als developer niet meer na te denken over hoeveel performance je app nodig heeft. De resources worden per verzoek gebruikt en als de app niet in gebruik is, zijn er ook geen resources allocated.

#### **Containers:**

- Een pakket van de software met alle dependencies zodat deze overal kunnen draaien en je geen zorgen moet maken over versies.

#### **lambda deployment package**

-

#### **concurrent executions:**

- Aantal requests die je tegelijk uitvoert.

#### **lambda image resizer:**

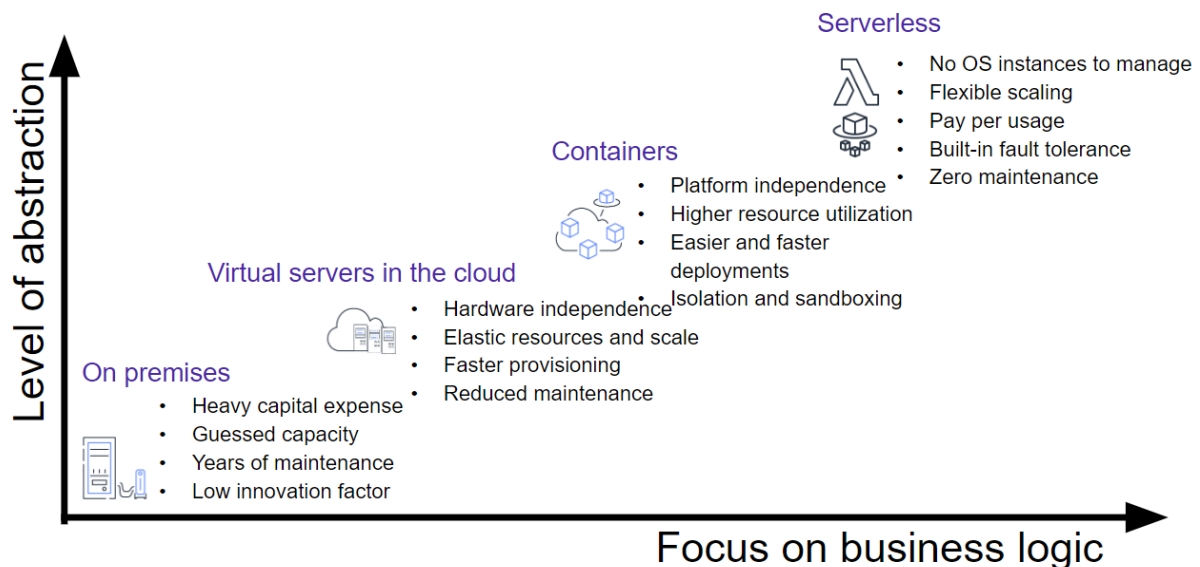
- Lambda kan foto's die op je S3 zijn opgeslagen resizen bij een request en slaat die dan op in de S3. Op deze manier hoef je niet 10 formaten van elke foto op voorhand te maken.

#### **API key: (application programming interface key)**

- Is een unieke identificatie methode om een gebruiker of programma te authenticeren en autorizeren die gebruik maakt van een API.

- kan in eigen woorden de evolutie naar serverless omschrijven en toelichten aan de hand van voorbeelden.

We zijn begonnen bij On premise (zelf eigen servers) met alle nadelen zoals hoge kosten en moeilijk te schalen naar de cloud. Hiermee valt het maintenance deel al weg en kun je veel makkelijker scalen als dit nodig is. Vervolgens werden de applicaties in containers gestoken waarmee je makkelijker nieuwe versies kunt uitbrengen en op elk OS je app kunt draaien. Ten slotte kwam Serverless die het OS en onderhoud weghalen.



- kan beschrijven welk de 4 key features zijn van een serverless world.

No servers to provision or manage

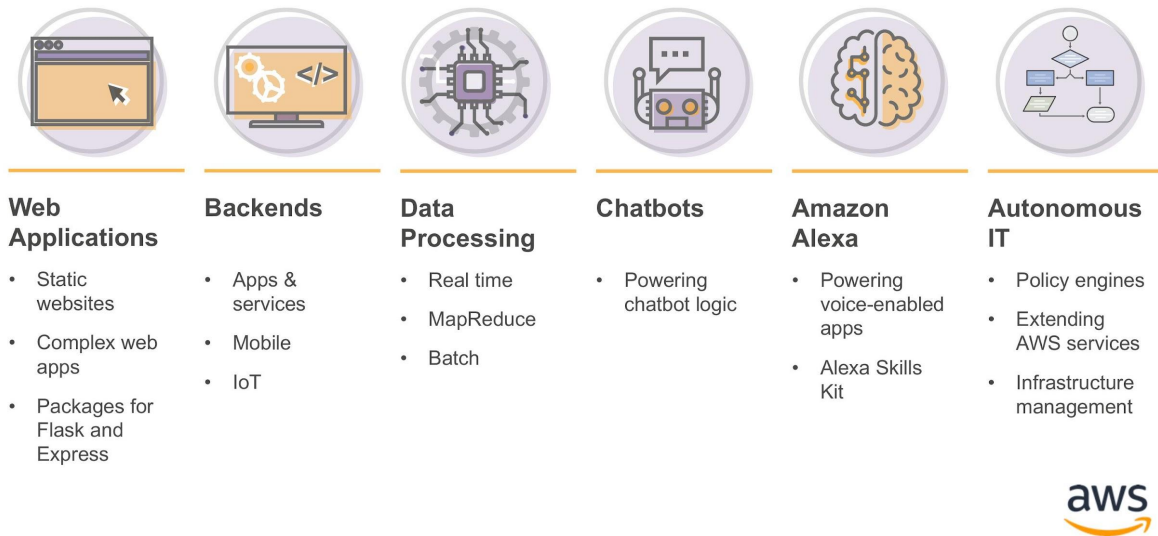
Scales with usage

Never pay for idle

Availability and fault tolerance built in

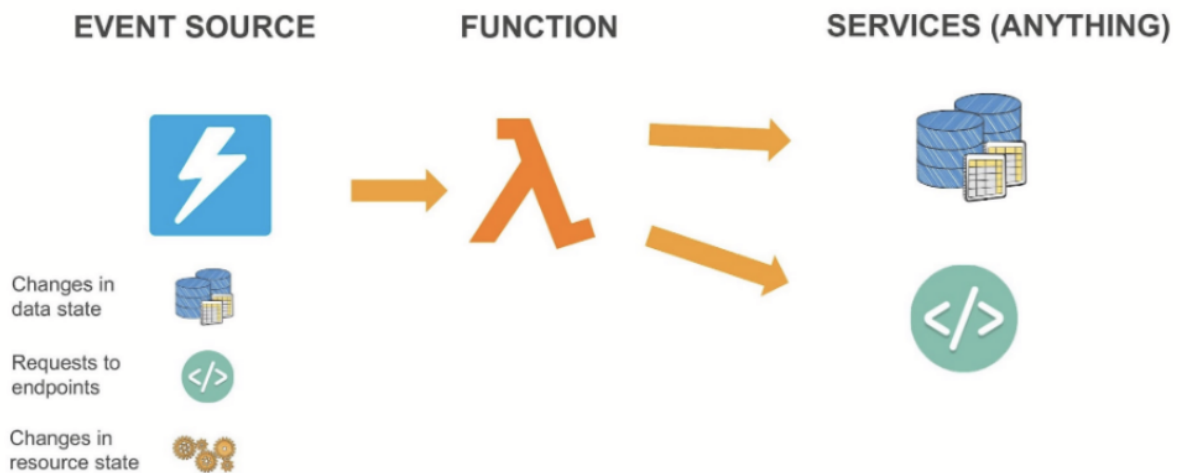
- kan specifieke use cases voor serverless applicaties aanhalen en in eigen woorden toelichten.

Verschillende use cases zoals Web applications, Backends,...



- kan de lambda workflow in eigen woorden omschrijven en toelichten

Eerst komt er een source, Dit kan een request of change in een data of resource state zijn. Dit roept Dit triggered dan de lambda functie die dan de services aanspreekt. Dit kan gaan van een S3 bucket naar databases naar ...



- kan minstens 2 event sources voor een lambda functie aanhalen en toelichten aan de hand van een use case

Changes in data state:

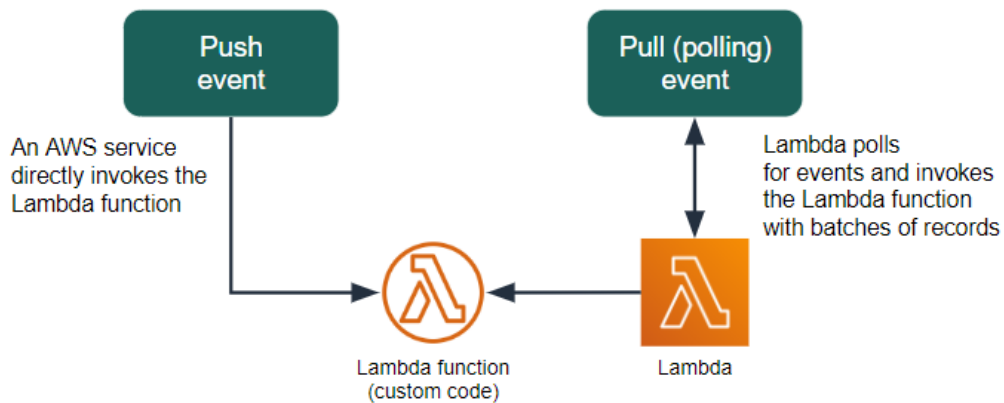
-

Changes in resource state:

-

Request to endpoints:

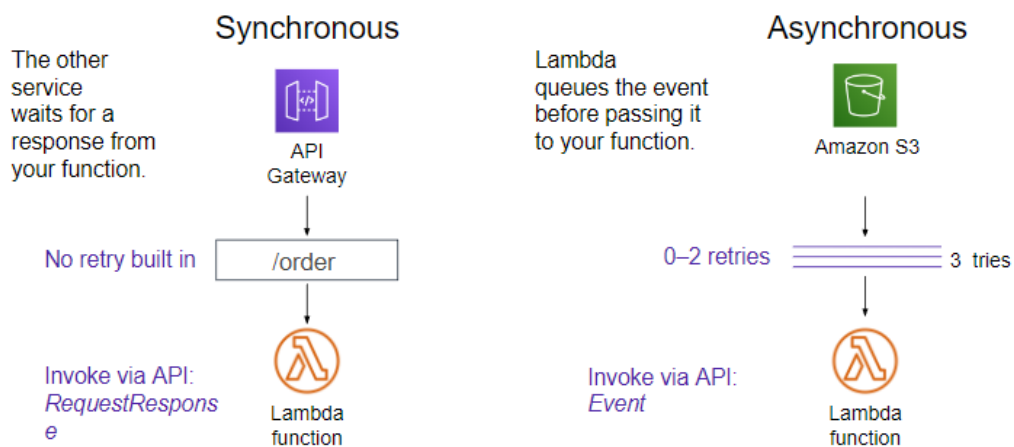
- Request naar een API gateway. (Zie PE2)
- kan het lambda execution model uitleggen, kan elke methode benoemen en deze linken aan een praktische case.



## Push event

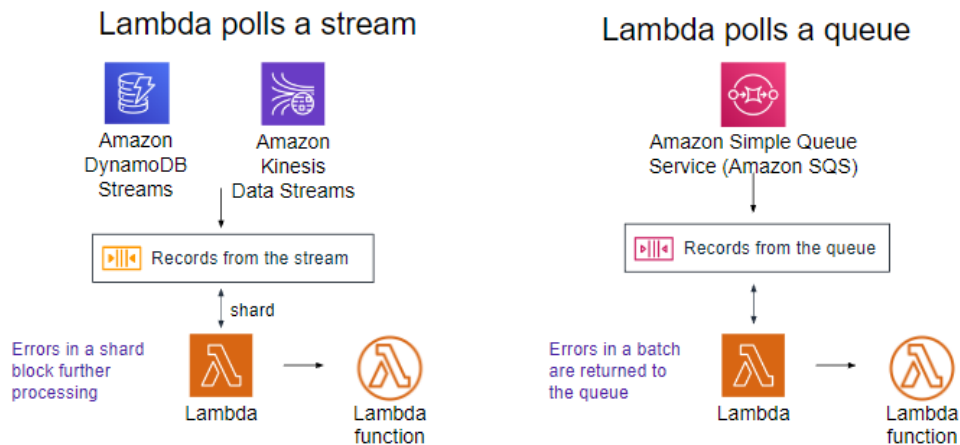
Voorbeeld: Een push naar een API gateway of iemand die een request maakt.

- **Synchroon:**
  - De service wacht op een response van je functie.
- **Asynchroon**
  - Lambda maakt een queue voor deze naar de lambda verstuurd worden



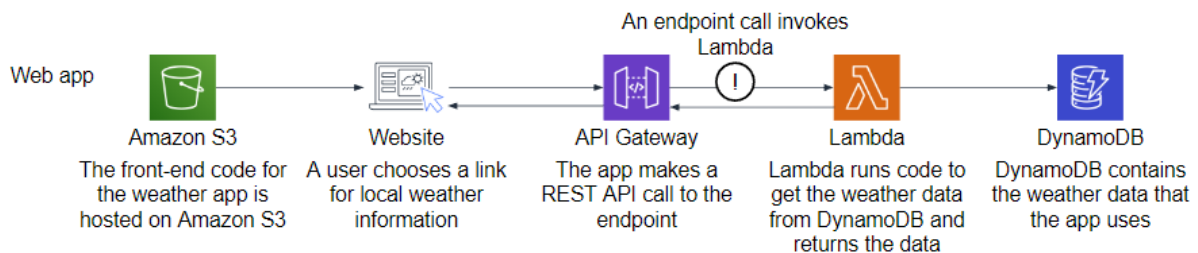
## Pull event

Voorbeeld: De lambda kijkt of er iets veranderd of nieuws is in je database en gaat dan de functie uitvoeren.



- kan voorbeelden geven van lambda workflows en kan deze linken aan een praktische case.

Voorbeeld: Denk aan PE



- kan de limitaties van het lambda resource- en deployment model toelichten en deze kaderen in een praktisch voorbeeld.
- kan de concepten in dit hoofdstuk linken aan elkaar en in eigen woorden de voordelen van serverless applicatie setups toelichten.



## 8. Services - API Gateway

### Doelstellingen

De student:

- kan de concepten Incident Response, AWS Certificate Manager, AWS Shield, AWS WAF, AWS SSO, AWS Directory service, data plane, control plane, Principle-of-Least-Privilege, MFA, AWS ARN, ... in eigen woorden uitleggen en kan hiervan concrete voorbeelden geven.

#### **Incident Response:**

- Georganiseerde aanpak van een organisatie om te reageren op beveiligingsincidenten en de impact ervan te minimaliseren.

#### **AWS Certificate Manager:**

- Beheerde service voor het eenvoudig implementeren en beheren van SSL/TLS-certificaten voor gebruik met AWS-services.

#### **AWS Shield:**

- Beheerde dienst voor DDoS-bescherming die AWS-infrastructuur en applicaties beschermt tegen aanvallen.

#### **AWS WAF:**

- Beheerde firewall-service voor het beschermen van webapplicaties tegen bekende webaanvallen.

#### **AWS SSO:**

- Beheerde service voor het centraliseren van toegangsbeheer en authenticatie voor meerdere AWS-accounts en bedrijfstoeepassingen.

#### **AWS Directory Service:**

- Beheerde service voor het opzetten en beheren van een directory om gebruikers, groepen en toegangsrechten centraal te beheren.

#### **Data plane:**

- Het deel van een netwerkkapparaat of service dat verantwoordelijk is voor het verwerken en doorsturen van netwerkverkeer.

#### **Control plane:**

- Het deel van een netwerkkapparaat of service dat verantwoordelijk is voor het beheer, de controle en het configureren van het apparaat of de service.

#### **Principle-of-Least-Privilege:**

- Beveiligingsprincipe dat aangeeft dat gebruikers alleen de minimale toegangsrechten moeten hebben die nodig zijn om hun taken uit te voeren.

**MFA (Multi-Factor Authentication):**

- Beveiligingsmethode waarbij gebruikers hun identiteit moeten verifiëren via meerdere onafhankelijke factoren, zoals een wachtwoord en een eenmalige verificatiecode.

**AWS ARN (Amazon Resource Name):**

- Unieke identificatie van een AWS-resource, zoals een EC2-instance, S3-bucket of IAM-gebruiker. Het wordt gebruikt voor het identificeren en toegangsbeheer van resources binnen AWS.
- 
- kan minstens 2 voordelen aanhalen van het gebruiken van API Gateway cache.

**Verbeterde prestaties:**

- Snellere respons en hogere doorvoer door het serveren van veelgevraagde API-responses vanuit de cache.

**Lagere latentie:**

- Vermindert de tijd die nodig is om een aanvraag te verwerken door de respons direct uit de cache te halen.

**Kostenoptimalisatie:**

- Verlaagt de belasting van backend-servers, waardoor kosten worden bespaard op serverresources en bandbreedte.

**Schaalbaarheid:**

- De cache verbetert de schaalbaarheid van de API door de belasting op de backend te verminderen en de verwerkingscapaciteit te vergroten.
- 
- kan de configuratie eigenschappen van de API Gateway benoemen en toelichten aan de hand van een voorbeeld

**API Resources**

- Resources vertegenwoordigen de paden van de API zoals /users, /products

**API Methods**

- Voor elke endpoint (resource) definieer je de bijbehorende methoden zoals GET, POST, PUT, DELETE, ...

- kan de AWS API Gateway workflow uitleggen in eigen woorden en de elementen toelichten aan de hand van voorbeelden.
- kan API deployments toelichten in eigen woorden.
- kan toelichten wat de voordelen zijn van een eigen/legacy API te encapsuleren in een API gateway.

## 9. Monitoring & Autoscaling

### Doelstellingen

De student:

- kan de concepten Metrics, Monitoring Agent, Amazon SNS alert, Amazon SQS, Namespace, Cloudtrail, Launch configuration, Manual scaling, scheduled scaling, Dynamic Scaling, Predictive Scaling, ... in eigen woorden uitleggen en kan hiervan concrete voorbeelden geven.

#### **Metrics**

- Metrics zijn meetbare gegevens die worden gebruikt om kwantitatieve informatie te verkrijgen over prestaties, gedrag en effectiviteit.

#### **Monitoring Agent**

- Een softwarecomponent die prestatie- en gedragsgegevens verzamelt van een systeem.

#### **Amazon SNS alert**

- Een meldingsservice voor het verzenden van waarschuwingen en meldingen op basis van vooraf gedefinieerde voorwaarden.

#### **Amazon SQS**

- Een beheerde berichtenservice voor het verzenden en verwerken van berichten tussen gedistribueerde systemen.

#### **Namespace**

- Een container of context voor het scheiden en organiseren van unieke namen en resources.

#### **Cloudtrail**

- Een service voor het bijhouden en controleren van API-activiteiten en gebeurtenissen in AWS.

#### **Launch configuration**

- Een sjabloon voor het configureren van bronnen en instellingen bij het starten van AWS EC2-instanties.

#### **Manual scaling**

- Handmatige aanpassing van de capaciteit van een systeem of resource.

#### **Scheduled scaling**

- Automatische aanpassing van de capaciteit op basis van een vooraf ingesteld schema.

### Dynamic scaling

- Automatische aanpassing van de capaciteit op basis van realtime werkbelasting en vraag.

### Predictive scaling

- Automatische aanpassing van de capaciteit op basis van voorspellende analyse van werkbelastingpatronen.
- Kan de verschillen tussen Event-based- en Time-based Events uitleggen en toelichten aan de hand van een eigen gekozen voorbeeld

**Event-based events:** gebeurtenissen die worden geactiveerd door specifieke gebeurtenissen of acties, zoals een gebruikersactie of een inkomend bericht.

**Voorbeeld:** Een melding naar een klant sturen wanneer een product weer op voorraad is.

**Time-based events:** gebeurtenissen die worden geactiveerd op basis van een vooraf ingesteld tijdsinterval of schema.

**Voorbeeld:** Het uitvoeren van een geautomatiseerde back-up van gegevens elke nacht om 02:00 uur.

- Kan het Cloudwatch metric model uitleggen in eigen woorden, alle componenten benoemen en hun functie toelichten in eigen woorden of aan de hand van een eigen gekozen voorbeeld.

Metric	Name and Value
Namespace	Group related metrics together
Dimensions	Name-value pairs that further categorize metrics
	Example: InstanceId a dimension of CPUUtilization
	Metric Name + Dimension = a new, unique metric
Period	Specified time (in seconds) over which metric was collected

- Kan toelichten hoe metric specifics toegepast worden om timeseries te creëren en hoe Dimensions toelaten om nieuwe, unieke metrics te creëren.

**Metric Specifics voor Timeseries:** Metriekspecificaties worden gebruikt om tijdsreeksen te creëren in CloudWatch. Ze omvatten de metrieknaam, waarde en

tijdstempel. Dit maakt het mogelijk om metriekwaarden te associëren met specifieke tijdstippen, waardoor prestaties in de tijd kunnen worden gevolgd en geanalyseerd.

**Dimensions voor unieke metrieken:** Dimensions maken het mogelijk om unieke metrieken te creëren. Ze voegen context en filtermogelijkheden toe aan metrieken. Door dimensions zoals resource-identificatie, locatie of eigenschappen toe te voegen, kun je specifieke aspecten van een systeem of service monitoren en unieke metrieken maken. Dit helpt bij het fijner afstemmen en analyseren van metrieken voor specifieke use-cases of resources.

- Kan de eigenschappen van Standard Metrics binnen Cloudwatch herkennen, benoemen en toepassen.
  - **Naam**
  - **Bron**
  - **Aggregatie-interval**
    - Tijdsinterval waarop de meting wordt uitgevoerd
- Cloudwatch Alarms zelf aanmaken, beheren en troubleshooten en kan de elementen die nodig zijn om een alarm te maken benoemen en uitleggen aan de hand van een voorbeeld.
  - Metric waarop je het alarm wilt baseren
  - Drempelwaardes voor het alarm
  - Optionele instellingen
    - Hoeveel keer grenswaarde overschrijden voor het in alarm gaat
    - max, min of avg

- Kan autoscaling aan de hand van een eigen gekozen voorbeeld uitleggen, kan de noodzakelijke componenten benoemen en toelichten en interpreteren.
  1. **Launch Configuration of Launch Template:** Bevat configuratiegegevens voor het starten van nieuwe EC2-instanties.
  2. **Auto Scaling Group:** Beheert en schaaft de instanties op basis van de configuratie.
  3. **Scaling Policies:** Definiëren de schalingsregels op basis van vooraf gedefinieerde criteria.
  4. **CloudWatch Alarms:** Monitoren de metrieken en activeren schalen op basis van alarmdrempels.
  5. **Load Balancer (optioneel):** Verdeelt het verkeer over de geschaalde instanties.