

# **Aurora Analyse Sprint 2**

Sprint 2

Auteur: Rahmi Tas

Datum: 5-11-2025

## A1. Requirements en compatibiliteit

Om te bepalen of Amazon Aurora geschikt is voor boomtag, heb ik onderzocht en gekeken welke database vereisten October CMS stelt en in hoeverre Aurora hieraan voldoet. De minimale vereiste versie is 5.7 maar versie 8.0 heeft de voorkeur aangezien daar beter prestaties bevinden, en ook stabiliteit en langere termijnondersteuning voorkomen. Daarnaast is er ook nog Aurora MySQL 3 dat volledig compatibel is met MySQL 8.0, op deze manier voldoet het daarmee aan alle bijbehorende functionele en technische eisen.

### Belangrijke aandachtspunten:

- PHP 8.2 met mysqli- en pdo\_mysql-extensies ondersteunt Aurora direct.
- Standaardkarakterset: utf8mb4 met collatie utf8mb4\_unicode\_ci.
- Bij oudere MySQL-versies kan een beperking optreden bij indexlengte (niet van toepassing bij Aurora 8.0).
- De projectcode volgt Laravel-conventies, waardoor migraties en tabellen direct compatibel zijn.

## A2. Architectuurvarianten

Om te bepalen welke Aurora-configuratie het meeste geschikt is voor Boomtag, zijn er drie verschillende architectuur varianten onderzocht. Het doel is om met dit analyse inzicht te krijgen in de balans tussen kosten, schaalbaarheid en beheercomplexiteit zodat de juiste keuzes gemaakt kan worden zowel ontwikkel als productie omgeving.

Aspect	Serverless v2 + Standard	Serverless v2 + I/O-Optimized	Provisioned + Standard
Schaalbaarheid	Automatisch 0.5–128 ACU	Automatisch 0.5–128 ACU	Handmatig (vertical/horizontal)
Kostenmodel	ACU + I/O + opslag	ACU + opslag (geen I/O-kosten)	Uurprijs + opslag + I/O
Gebruik	Dev/test met lage load	Kleine productie met veel I/O	Constance productie-load
Beheer	Weinig onderhoud	Weinig onderhoud	Meer handmatig beheer

Voordeel	Laagste kosten bij lage load	Voordeliger bij veel I/O	Stabiel bij vaste load
----------	------------------------------	--------------------------	------------------------

#### Analyse per variant:

- *Serverless v2 + standard*: Dit is ideaal voor ontwikkel en testomgeving met continu wisselende belasting. Hierbij zijn de kosten ook nog laag zolang de database niet continu actief is en als plus punt schaaft het systeem automatisch bij piek momenten.
- *Serverless v2 + I/O Optimized*: Dit is meer geschikt voor productie omgeving met veel lees en schrijfbewerkingen. Hoewel de opslag en de kosten hiervan duurder is, worden er geen I/O kosten berekend wat het voordeliger maakt bij intensief gebruik.
- *Provisioned + standard*: Dit biedt voorspelbare prestaties voor omgeving met een stabiel verkeer en niet iets met continu pieken. Als min punt mist het wel de flexibiliteit en kost efficiëntie van de serverless varianten.

## A3. Betrouwbaarheid en herstel

Aurora is ontworpen voor hoge beschikbaarheid en automatische fouttolerantie. De database herhaalt data over drie verschillende zones binnen een AWS regio waardoor de kans op data verlies of downtime minimaal is. Daarnaast is het zo dat wanneer er een zone uitvalt, blijft het systeem nog steeds operationeel.

Voor Boomtag is het opslaan van data en het netjes hebben belangrijk. Er moet wel nog gekeken worden naar de kosten aangezien die in verhouding moeten blijven met de ontwikkelfase met dat wordt in A6 besproken. Daarom is er gekozen voor een herstelconcept dat de juiste balans biedt tussen beschikbaarheid, herstelling en kosten.

- RTO (Recovery Time Objective): maximaal 30 minuten, zodat de database binnen een half uur hersteld kan worden bij een incident.
- RPO (Recovery Point Objective): maximaal 5 minuten, wat betekent dat er maar hoogstens vijf minuten aan data kwijt kan raken.
- Dagelijkse automatische back-ups, 7 dagen bewaard in dev en 14 dagen in prod.
- Handmatige snapshots: Dit is gemaakt voor en na releases, om snel terug te keren naar een stabiele herstellpunt
- Point-in-Time Recovery: ingeschakeld voor herstel op specifieke tijdstippen, voornamelijk op minuutniveau

Instelling	Dev	Prod	Effect
Backupretentie	7 dagen	14 dagen	Langer herstelvenster
Snapshots	Handmatig	Pre-release	Snelle rollback
PITR	Aan	Aan	Herstel tot specifiek moment
Multi-AZ	Aan	Aan	Hoge beschikbaarheid

## A4. Security en netwerk

De beveiliging van de aurora database is van cruciaal belang omdat hierin gebruikersgegevens en belangrijke data van boomtag in kunnen zitten. Het ontwerp hiervan richt zich op netwerkisolatie, toegangsbeveiliging en geheimbeheer zodat de kans op verkeerde configuratie en rechten minimaal blijft. Hieronder zie je een lijst van belangrijke maatregelen en een korte onderbouwing ervan:

- **Private subnets:** Aurora wordt uitsluitend in private subnets geplaatst, zonder enige publieke endpoint. Hierdoor is directe toegang ertot vanuit het internet compleet onmogelijk. Dit vermindert het aanvalsoppervlak.
- **Versleutelde verbindingen:** Alle database verbindingen verlopen via TLS, en de data hiervan wordt opgeslagen met AWS Key Management Service (KMS). Dit voorkomt datalekken bij netwerk en opslag incidenten.
- **Security groups:** Alleen de applicatieserver krijgt toegang tot poort 3306. De rest van de inkomende verbindingen worden automatisch geweigerd, wat het risico op brute force minimaliseert.
- **Secrets Manager:** Hierin worden wachtwoorden en connectiestrings opgeslagen in AWS Secret Manager. Dit voorkomt menselijke fouten en vermindert het risico op het lekken van de credentials.
- **Least-privilege IAM:** IAM rollen worden specifiek ingericht voor de database, terraform en CI/CD. Zo heeft elke component alleen de rechten die bij hun van spraken zijn.
- **Infrastructuur via Terraform:** Alle instellingen qua security worden vastgesteld, waardoor configuratie fouten automatisch kunnen worden gedetecteerd.

- **Audit en Monitoring:** AWS Cloudwatch en AWS Config monitoren wijzigingen en loggen de toeganspogingen, wat detectie van verdachte activiteiten mogelijk maakt.

## A5. Parameters en observability

### Aanbevolen parameters en motivatie

Voor optimale prestatie en inzicht in de werking van Aurora zijn specifieke instellingen binnen de parameter groep aanbevolen. Deze parameters zorgen ervoor dat de database compatibel kan blijven met de October CMS, goed omgaat met transacties en goed kan worden bekeken in productie.

- `character_set_server` = `utf8mb4`
- `collation_server` = `utf8mb4_unicode_ci`
- `performance_schema` = 1 (alleen productie)
- `slow_query_log` = 1 met `long_query_time` = 0.5–1s
- `innodb_flush_log_at_trx_commit` = 1 (prod) / 2 (dev)
- `max_connections` afgestemd op de applicatie (100–200)

Met de instellingen van hierboven kan boomtag de balans bewaren tussen prestaties, betrouwbaarheid en inzicht in database gedrag zonder enige overbodige overhead.

### Observability en meetplan

Aurora biedt integratie met Amazon CloudWatch waarmee je de prestaties en betrouwbaarheid continu kan volgen. De volgende metrieken worden aanbevolen om in te stellen als dashboards en alarms:

Metriek	Doel	Waarschuwingdrempel
<b>CPUUtilization / ACUUsage</b>	Meet rekencapaciteit en schaalgedrag	> 80% gedurende > 5 minuten
<b>DatabaseConnections</b>	Detecteert verbindingspieken of leaks	> 90% van <code>max_connections</code>
<b>DiskQueueDepth / IOPS</b>	Meet I/O-druk op opslaglaag	> 300 IOPS consistent

<b>Deadlocks</b>	Signaleert concurrency-problemen	> 0 per minuut
<b>ReplicaLag</b>	Controleert synchronisatie tussen AZ's	> 1 seconde vertraging
<b>ErrorLogs / SlowQueries</b>	Detecteert prestatieproblemen	Stijging > 20% week-op-week

## A6. Kosteninschatting

Om inzicht te krijgen in de financiële impact van Aurora voor Boomtag, zijn twee realistische scenario's doorgerekend: een kleine ontwikkelomgeving en een lichte productie omgeving. De berekeningen zijn gebaseerd op gemiddelde AWS-tarieven voor de regio *eu-central-1 (Frankfurt)*

Scenario	Configuratie	Verwachte load	Geschatte maandkosten	Opmerkingen
<b>Dev klein</b>	Serverless v2 + Standard	0,5–1 ACU, 20 GB opslag, 5M I/O's	<b>\$65–80 / maand</b>	Laagste kosten; ideaal voor ontwikkeling en test.
<b>Prod klein</b>	Serverless v2 + I/O-Optimized	2–4 ACU, 50 GB opslag, 50M I/O's	<b>\$180–200 / maand</b>	Voordeliger bij hogere I/O-belasting.

### Belangrijkste aannames:

- Prijs per ACU uur is ongeveer \$0.12 (serverless v2)
- Opslag is ongeveer \$0,10 per gb/maand
- I/O verzoeken is ongeveer \$0.20 per miljoen (alleen bij de standard)
- I/O Optimized reken geen I/O verzoeken maar de opslag hiervan is wel 25% duurder

## **Korte Analyse plus conclusie:**

Bij lege of niet veel voorkomende belasting is *Serverless v2 + standard* financieel het meest aantrekkelijk. De I/O kosten blijven dan namelijk beperkt en Aurora kan dan automatisch schalen zonder enige onnodige ACU verbruik. Om wel mee te nemen is zodra het platform echt structureel meer dan 30 tot 40 miljoen I/O's per maand verwerkt, wordt I/O Optimized voordeliger. Dit geldt normaal gesproken voor de productie omgeving met veel gebruikers die tegelijkertijd bezig zijn.

## **BRONNEN**

*-Requirements en Compabiliteit*

### **AWS Documentation – Aurora MySQL version 8.0 and 5.7**

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.CompareMySQL57.html>

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.MySQL80.html>

*-A2 Architectuurvarianten*

### **AWS Documentation – Aurora Serverless v2 Overview**

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-serverless-v2.html>

### **AWS Documentation – Aurora Storage and I/O-Optimized configuration**

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.StorageReliability.html>

*-A3 betrouwbaarheid en herstel*

### **AWS Documentation – Amazon Aurora High Availability and Durability**

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Concepts.AuroraHighAvailability.html>

### **AWS Documentation – Backing Up and Restoring an Aurora DB Cluster**

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/BackupRestoreAurora.html>

### **AWS Documentation – Aurora Global Database and Recovery Objectives**

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-global-database-disaster-recovery.html>

#### *-A4 Security en network*

Geen gebruik gemaakt van bronnen, onderzocht en gekeken naar eigen structuur en toegepaste kennis van vorige sprint.

#### *-A5 Parameters en observability*

### **AWS Documentation – Monitoring Amazon Aurora with Amazon CloudWatch**

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/MonitoringAurora.html>

### **AWS Documentation – Amazon CloudWatch Metrics and Alarms**

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html>

### **AWS Documentation – Amazon Aurora Parameter Groups**

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Reference.ParameterGroups.html>

#### *-A6 Kosteninschatting*

### **AWS Pricing – Amazon Aurora**

<https://aws.amazon.com/rds/aurora/pricing/>

### **GitLab Documentation – Security and Compliance Scanning in CI/CD**

[https://docs.gitlab.com/ee/user/application\\_security/iac\\_scanning/](https://docs.gitlab.com/ee/user/application_security/iac_scanning/)