ANALYSE DOCUMENT AWS

Groep

Alan Dhooge,

Ignace Shoeib,

Kaine Van Geffen,

Massin Lajlufi

AP Hogeschool

ICT Architecture

[1 - SAMENVATTING 2](#_Toc89793000)

[2 – SITUATIE-TO-BE 3](#_Toc89793001)

[3 - ARCHITECTUUR 4](#_Toc89793002)

[4 - VERANTWOORDING KEUZES 5](#_Toc89793003)

# 1 – SAMENVATTING

De gebruikers moeten bestanden kunnen delen met andere gebruikers.

Bij deze zal de server een UUID genereren voor elk bestand die de gebruiker upload. Als de gebruiker dan deze bestand heeft geüpload, dan zal de server een checksum moeten uitvoeren. Een checksum is een som gebaseerd op het aantal bits die een bestand heeft, de reden dat we dan een checksum willen is om errors te voorkomen.

Aangezien we gebruikers moeten aanmaken, zullen we gebruik maken van **Amazon Cognito**

Amazon Cognito is verantwoordelijk voor het mogelijk maken om gebruikers toe te voegen en te kunnen authentiseren via onze API. Het is ook mogelijk om deze gebruikers eigen rollen te geven.

Omdat de server een UUID genereert voor elk bestand die de gebruiker upload, zullen we deze UUID van de bestanden moeten bewaren in een **Amazon Relational Database**. We kunnen met deze UUID een checksum uitvoeren om te zien of deze bestand foutloos is overgebracht.

Als een gebruiker een bestand heeft geüpload, zal deze bestand(en) maar maximaal 24 uur op de server blijven. Het kost veel tijd om dit handmatig te doen, en is dus hierdoor ook niet efficiënt. Om dit te automatiseren zullen we gebruik maken van **Amazon Lambda.** Amazon Lambda zal op basis van onze zelf-ingestelde momenten automatisch onze S3 Bucket leegmaken.

# 2 – SITUATIE-TO-BE

**We zullen gebruik maken van de volgende Amazon Web Services:**

**Amazon S3**

**Amazon S3 is een service om objecten op te slaan. We gebruiker deze service omdat de bestanden die de gebruiker upload ergens moet bewaard worden.**

**Deze service heeft ook verbinding met een EC2 instantie om bestanden te delen met andere gebruikers.**

**Amazon EC2**

De backend van de applicatie zal zich hier bevinden. Amazon EC2 is verantwoordelijk voor het contact met het front-end van de applicatie en de S3 bucket. Deze service zorgt er ook voor dat je applicatie schaalbaar is.

**Amazon Cognito**

Amazon Cognito is verantwoordelijk voor het authenticatie van gebruikers, en toegang voorzien tussen applicaties en Amazon.

We zullen gebruik maken van deze service om ervoor te zorgen dat enkel geauthentiseerde gebruikers bestanden kunnen uploaden.

Om ervoor te zorgen dat geauthentiseerde gebruikers onderscheden kunnen worden, zullen we gebruik maken van een User Pool waarin de gebruiker zich kan registreren en inloggen via het front-end.

Alleen geauthentiseerde gebruikers kunnen uploaden en downloaden.

**Amazon RDS**

**[ZEKER NOG AANPASSEN]** We willen dat de informatie van de bestanden ook apart bewaard wordt in een database. Zodra een gebruiker een bestand uploadt zullen de gegevens van deze bestand (FileID & FileName) in een database geplaatst worden. Onze database zal werken aan de hand van Mysql. Als een gebruiker een bestand uploadt zal er ook een Mysql query uitgevoerd worden die de metadata van het bestand toevoegt aan de tabellen.

**Amazon Lambda**

Amazon Lambda is een manier om automatisch een taak te laten uitvoeren zonder manuele interactie van de gebruiker.

We maken gebruik van Amazon Lambda om ervoor te zorgen dat alle objecten (bestanden) binnen onze Amazon S3 bucket automatisch na 24 uur verwijderd worden.We willen namelijk, volgens de opdracht, dat bestanden niet langer dan 1 op onze systeem bewaard worden.

Hetzelfde moet gebeuren in onze MySQL AWS Relational Database, Als de toegevoegde record van onze bestand ouder is dan 24 uur, moet deze record verwijderd worden van de database.

**Interface / API**

We maken gebruik van de Swagger API als frontend. Met deze API kun je eventuele informatie per controller ingeven.

Deze informatie word verwerkt in de code en naar de corresponderende Service gestuurd.

Onze controllers folder bevat het volgende:

* **FilesController**

***Het uploaden van bestanden***

De gebruiker geeft een bestand in als input. Deze file word gebruikt als parameter in de functie om bestanden te uploaden.

We krijgen hierbij onze UUID en file terug indien deze succesvol is uitgevoerd.

***Het downloaden van bestanden***

De gebruiker geeft een string waarde in, deze word dan gezocht in de bucket.

Als deze bestand is gevonden, zullen we **[UHH IDK]** ontvangen

* **UserController**

***Het aanmaken van een gebruiker***

De gebruiker moet een email (string) en wachtwoord (string) ingeven. Deze informatie word verzonden naar de Amazon Cognito service door een SignUpRequest te creëren.

De gebruiker zal nadien aangemaakt en beschikbaar gesteld zijn in de Amazon Cognito service.

***Het authenticeren van een bestaande gebruiker***

De gebruiker moet een email (string) en wachtwoord (string) aangeven.

Er word met deze informatie een authentication request aangevraagd aan de Amazon Cognito service.

Als deze succesvol is, **[AANVULLEN]**

Onze **Models??** Folder bevat het volgende:

* **AWSCredentials**

Deze klasse is een hulpmiddel, verantwoordelijk om makkelijker een verbinding te maken met Amazon Web Services.

# 3 – ARCHITECTUUR

**Back-end**

* C# (ASP .NET)
* Python (lambda functie)
* MySQL (databases)

ASP.NET zal het mogelijk voor ons maken om makkelijk te communiceren met onze Amazon Web Services.

**Controllers**

Controllers maken het mogelijk om HTTP requests op te vangen, te versturen en bij te werken. In ons programma hebben we de volgende Controllers

* **FilesController**

De Files Controller zorgt ervoor dat de gebruiker bestanden kan uploaden, en bestanden kan opvragen

* **RDSController**

De RDSController zorgt ervoor dat we kunnen communiceren met de Database van Amazon Web Services, Er zal een query uitgevoerd worden om informatie te plaatsen binnen de Files database.

* **UserController**

De UserController is verantwoordelijk voor het aanmaken van een gebruiker in Amazon Cognito, en het authentiseren van een gebruiker

# 4 - VERANTWOORDING KEUZES

**Programmeertaal**

We hebben besloten om met C# te werken aangezien we meer kennis en ervaring hebben met deze programmertaal.

Om onze lambda functie te programmeren hebben we gebruik gemaakt van Python. Deze is beter compatibel met de editor die beschikbaar is in AWS zelf, en werkt ook eenvoudiger. Bij C# is dit helaas niet het geval.

We hebben ook gebruik gemaakt van MySQL, deze enkel om een query uit te voeren

**API**

We maken gebruik van Swagger.

Swagger is een soort template die de functies van onze .ASP applicatie weergeeft, dit is handig indien de gebruiker een gestructureerde, overzichtelijke interface heeft van zijn code.

De gebruiker kan hierbij zijn eigen informatie ingeven (Files, Tekst, Getallen, afhangend op welke datatype de parameter van de functie is ingesteld). Met deze informatie word de functie uitgevoerd waar de gebruiker deze informatie heeft ingegeven. De resultaat van deze functie word in een kader weergegeven onder de inputveld.

**Services**

We maken gebruik van:

* **Amazon S3**

Omdat we een opslagplaats willen voor onze bestanden. De functionaliteit van Amazon S3 (Opslagplaats voor objecten) is daarom het meest toepasselijke voor deze functie.

* **Amazon RDS**

We willen de informatie van onze bestanden behouden nadat de gebruiker deze upload. RDS is het meest geschikte om informatie te bewaren omdat het een database is.

* **Amazon Cognito**

We willen ervoor zorgen dat gebruikers zich moeten authenticeren, Cognito is precies voor deze doeleinden geschikt.