**De Cloud Resume Challenge**

Hi! Mijn naam is Tristan Graaff. Ik ben op dit moment werkzaam als Test Automation Engineer en doe dit werk met plezier. De laatste tijd ben ik echter steeds meer geïnteresseerd geraakt in cloud computing en tech-infrastructuur en overweeg ik om een carrièreswitch te maken naar Cloud Engineer. Wat lijkt me nou zo leuk aan het werk als Cloud Engineer? Dat kan ik met alle liefde uitleggen.

* Van nature ben ik een jack-of-all-trades. Ik heb veel interesses en haal er voldoening uit met divers bezig te zijn. Het werk van een Cloud Engineer bestaat uit vele facetten: operating systems, networking, security, infrastructure as code, databases, pipelines, en zo voort.. Hier kan ik al mijn IT-interesses in kwijt, de ene dag ben ik misschien wat IaC aan het coderen, terwijl ik de andere dag werk aan de netwerkbeveiliging.
* State-of-the-art technologie. Cloud computing is een van de ‘hottest’ IT-vakgebieden. Je komt te werken met de nieuwste technologieën en staat aan het voorfront van de nieuwste ontwikkelingen binnen de IT, en al helemaal als je bijvoorbeeld AI-modellen gaat trainen en beheren in de cloud.
* Carrièreperspectief. Cloud computing is op dit moment een van de meest gevraagde vakgebieden in de IT en in de toekomst zullen alleen maar meer bedrijven cloud-native gaan werken. Er is veel werk en er zijn veel groeimogelijkheden te vinden in deze sector, zelfs meer dan in andere IT-vakgebieden, daar ben ik van overtuigd.

Maar hoe kom je er na wat te lezen en wat video’s te kijken achter of een bepaald onderwerp je echt ligt? Door het te doen natuurlijk. Ik was op zoek naar doe-het-zelf projecten om over de cloud te leren, en kwam toen de Cloud Resume Challenge tegen. Ik heb besloten om deze aan te gaan om te kijken hoe ik het werken in de cloud zelf ervaar, en om te kunnen laten zien wat ik kan, mocht ik door willen gaan met mijn reis tot Cloud Engineer. Ik heb besloten te gaan werken met AWS, omdat dit wereldwijd de grootste cloud provider is, en omdat de look en feel mij het meest bevalt van de 3 grote providers (AWS, Azure en GCP). Het belangrijkst is om aan de slag te gaan en te beginnen met leren. Uiteindelijk zijn deze platformen slechts tools om problemen in de echte wereld op te lossen. Zie hier meer info over de Cloud Resume Challenge: <https://cloudresumechallenge.dev/docs/the-challenge/aws/>. De komende tijd zal ik documenteren wat ik heb geleerd en dat op mijn blog posten.

Tot snel!

**Het veilig hosten van een statische website**

De Cloud Resume Challenge (<https://cloudresumechallenge.dev/docs/the-challenge/aws/>) bestaat uit 16 beschreven stappen, maar eigenlijk zie ik hem in drie delen.

1. Het hosten van de website
2. Het toevoegen van dynamische functionaliteit aan de website
3. Het implementeren van DevOps principes zoals Infrastructure as Code en CI/CD pipelines

In deze blogpost laat ik zien hoe ik het hosten van de website heb gerealiseerd.

**Gebruikte services:**

* **AWS S3**
* **AWS CloudFront**
* **AWS Route 53**
* **AWS Certificate Manager (ACM)**

**Het aanmaken van S3 buckets**

Ik ben begonnen met het maken van een html pagina (die waar je nu op kijkt) en heb hier wat css voor de styling aan toegevoegd. Vervolgens heb ik 3 nieuwe buckets in Amazon S3 aangemaakt. Zoals je kan zien heb ik de regio eu-central-1 gekozen. Dit is de regio waar mijn potentiële websitebezoekers zicht het dichts bij bevinden.

A screenshot of a computer

AI-generated content may be incorrect.

We uploaden de content van de website naar de tristantech.org bucket. Dit kan met de AWS console, maar als je de AWS CLI hebt geïnstalleerd, kan dit ook met het volgende commando: *aws s3 cp <file-path> s3://tristantech.org.* Na het uitvoeren van dit commando zijn de upgeloade files zichtbaar in de bucket.

A screenshot of a computer

AI-generated content may be incorrect.

In de tristantech.org bucket heb ik de volgende instellingen aangepast:

* *Static website hosting* is enabled met de volgende instellingen. De homepage van de website wordt de index.html file, en we voegen ook een error document toe, mocht de website niet correct kunnen laden.

A screenshot of a computer

AI-generated content may be incorrect.

* *Block all public access* is enabled. Dit is een best practise voor security wanneer CloudFront wordt gebruikt.

A text on a white background

AI-generated content may be incorrect.

* We richten de volgende bucket policy in: A screen shot of a computer program

  AI-generated content may be incorrect.

Hiermee krijgt CloudFront expliciet wel toegang tot de bucket en zal de website dus alleen via deze route kunnen worden bezocht. Daarnaast is alle toegang verboden tot de folder */other*. Hier heb ik andere delen van mijn website staan die ik nog niet met het publiek wil delen.

Nu de volgende bucket. De www.tristantech.org bucket blijft leeg.

A screenshot of a computer

AI-generated content may be incorrect.

De reden hiervoor is dat deze enkel moet redirecten naar de tristantech.org bucket, waar onze content staat. Dit kan worden ingesteld onder Properties -> Static website hosting.  
  
A screenshot of a computer

AI-generated content may be incorrect.

En in de laatste bucket, logs.tristantech.org, zien we enkel een *logs* folder.

A screenshot of a computer

AI-generated content may be incorrect.

Deze wordt automatisch toegevoegd wanneer in de tristantech.org bucket de instelling *Server access logging* op enabled wordt gezet en de logs.tristantech.org bucket wordt gekozen om logging weg te schrijven.

A screenshot of a computer

AI-generated content may be incorrect.

**Het hosten van onze statische website op een veilige manier**

**Gebruikte services:**

* **AWS S3**
* **AWS CloudFront**
* **AWS Route 53**
* **AWS Certificate Manager (ACM)**

Nu de S3 buckets zijn aangemaakt, wordt het tijd om te gaan nadenken over een stukje architectuur. We gaan namelijk meerdere AWS services met elkaar laten communiceren. Laten we eerst high-level overview maken zodat duidelijk wordt welke service met welke communiceert en waarom.

A diagram of a diagram

AI-generated content may be incorrect.

Helemaal links zien we de user, de websitebezoeker. Deze zal het adres van de website intoetsen, tristantech.org, en verwacht vervolgens mijn website te zien. Dit is de route die wordt afgelegd, van user naar de content in de S3 bucket:

1. De user toetst een website in. Deze moet worden omgezet naar een IP-adres. AWS Route 53 is een DNS-service waarmee ik de domeinnaam koppel aan de juiste CloudFront-distributie. Dit betekent dat er DNS-servers worden geraadpleegt om het IP-adres te vinden dat bij het domein hoort.
2. Wanneer het IP-adres is gevonden, wordt het verzoek van de gebruiker doorgestuurd naar Amazon CloudFront, een content delivery network. CloudFront heeft zogeheten *edge locations* over de hele wereld, waardoor het content kan leveren vanaf een locatie die het dichtst bij de gebruiker ligt. Kopieën van de websitecontent worden in cache opgeslagen om deze zo snel mogelijk te kunnen leveren aan de eindgebruiker.
3. Gezien we een veilige verbinding willen opstellen, willen we gebruik maken van HTTPS zodat het verkeer tussen de webbrowser van de gebruiker en onze website versleuteld is. Hiervoor is een SSL/TLS certificaat nodig dat kan worden uitgegeven en beheerd door Amazon Certificate Manager (ACM). Wanneer we een geldig certificaat is, kan HTTPS worden ingeschakeld in CloudFront.
4. Uiteindelijk haalt CloudFront de statische content van de website op uit de S3 bucket die ik in de vorige blogpost heb aangemaakt en waar ik mijn websitecontent heb opgeslagen. CloudFront zal de content eerst uit zijn cache proberen te laden, en indien de content hier niet aanwezig is, dan zal de bucket worden geraadpleegd.