**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 een 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, oftewel 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 mijn CloudFront-distributie. Er worden DNS-servers geraadpleegd 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 hebben opgevraagd en gekoppeld, 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 bij ieder nieuw websitebezoek de content eerst uit de cache proberen te laden, en indien de content hier niet aanwezig is, dan zal de bucket worden geraadpleegd.

Laten we nu de services een voor een gaan opzetten.

**Route 53**

De eerste stap is om een domein aan te schaffen, als je dit nog niet bezit. Dit kan via Amazon, maar ook via andere aanbieders zoals GoDaddy. Ik heb mijn domein via Amazon gekocht. Zie hieronder hoe je een domein aanschaft (ik gebruik hier een voorbeeldnaam omdat ik tristantech.org al heb gekocht).

A screenshot of a computer

AI-generated content may be incorrect.

Na het aanschaffen zal het gekozen domein in *Route 53 -> Registered domains*  verschijnen.

A screenshot of a computer

AI-generated content may be incorrect.

Vervolgens heb ik in *Route 53 -> Hosted zones -> Create hosted zone* een nieuwe hosted zone gecreëerd. In een hosted zone kunnen meerdere records worden beheerd die onder hetzelfde domein vallen, hier zal ik zo op terugkomen om dit in praktijk te laten zien hoe dit werkt.

A screenshot of a computer

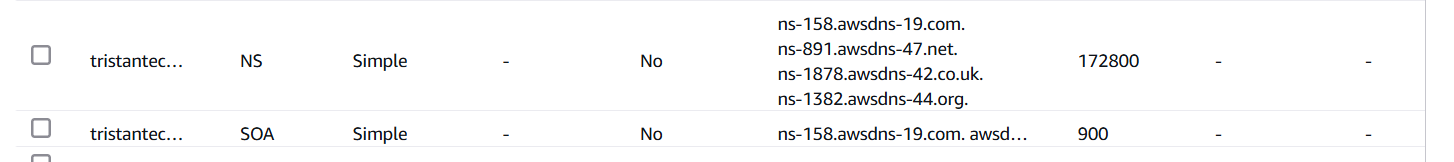
AI-generated content may be incorrect.

De hosted zone is nu zichtbaar onder het overzicht *Hosted zones*.

A screenshot of a computer

AI-generated content may be incorrect.

Bij het openen van de aangemaakte hosted zone, zijn er twee records automatisch aangemaakt.



Het name server (NS) record bevat 4 name servers. Dit zijn de servers die de DNS-queries naar mijn domein afhandelen. Deze name servers moeten gelijk zijn aan de name servers in het geregistreerde domein. Route 53 zal in de eerste instantie dezelfde name servers toewijzen, maar mocht je ze ooit moeten wijzigen, dan kan je dit bij *Registered domains* doen (dus niet in de hosted zone zelf).

A close-up of a computer screen

AI-generated content may be incorrect.

Dan is er nog het start of authority (SOA) record. Hierin staat belangrijke informatie over de hosted zone, zoals de primaire name server en het emailadres van de administrator van het domein. Zowel de gegevens van de NS als de SOA zijn publiek beschikbaar. Met een simpel PowerShell command kan ik ze opvragen.

A screenshot of a computer screen

AI-generated content may be incorrect.

**ACM**

Nu is het tijd om een certificaat aan te vragen in de AWS Certificate Manager (ACM). Let op! Ik werk vanuit de AWS region eu-central-1. Het certificaat moet echter vanuit us-east-1 worden aangevraagd, anders zal het niet worden goedgekeurd.

A screenshot of a computer

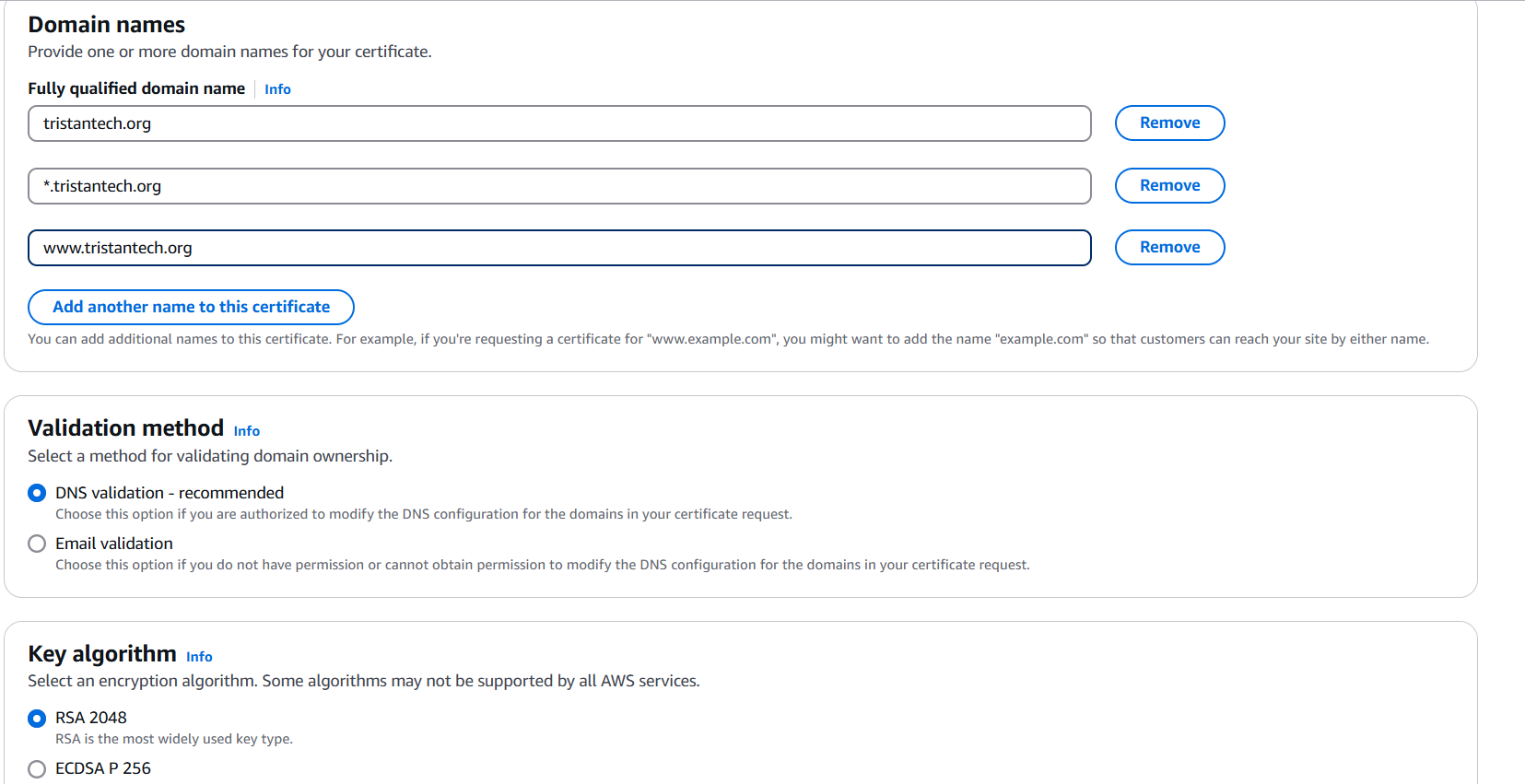
AI-generated content may be incorrect.

Ik vraag een public certificate aan.

A screenshot of a computer

AI-generated content may be incorrect.

Ik voer mijn domeinnaam in, en daarnaast voeg ik ook twee subdomeinen toe, en kies verder voor de standaardinstellingen. Dat betekent dat we ons certificaat gaan valideren door middel van de DNS.

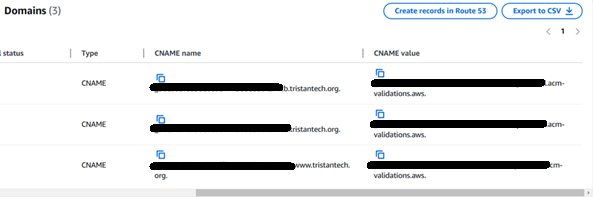


Nu het certificaat is aangemaakt, kan ik het openen, maar is de status nog *pending*. (Hieronder is een voorbeeldcertificaat te zien, niet het werkelijke dat ik gebruik).

A screenshot of a computer

AI-generated content may be incorrect.

In het certificaat staan onder *Domains* mijn domeinen met een CNAME name en een CNAME value. Deze waarden heb ik zo nodig in de *Route 53 Hosted zone* om het certificaat te valideren, dit is de DNS validatie. Zoals je ziet zijn twee CNAME names en values gelijk aan elkaar. Dit is omdat dat domeinen tristantech.org en \*.tristantech.org dezelfde names en values toegewezen krijgen. Het is dus niet nodig om ook een \*.tristantech.org aan te maken.



**Terug naar Route 53**

Nu maak ik in *Route 53 -> Hosted zones -> tristantech.org -> Create record* voor beide bovenstaande domeinen een nieuw CNAME record aan.

A screenshot of a computer

AI-generated content may be incorrect.

Nu ga ik terug naar mijn certificaat en de status is nu *Issued* in plaats van *Pending*. Het certificaat is gevalideerd op mijn domein. Hiermee heb ik bewezen dat ik de eigenaar ben van het domein.

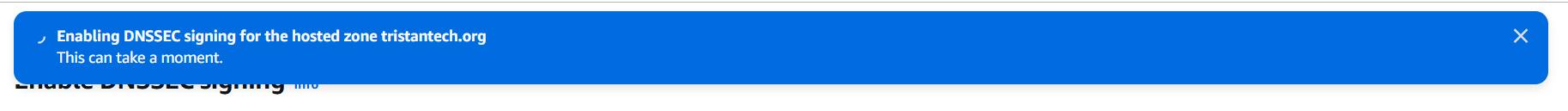
Wat ik ook nog wil doen, is het implementeren van DNSSEC in mijn hosted zone. Met DNSSEC worden cryptografische handtekeningen toegevoegd aan de records in mijn hosted zone. Door de handtekening te controleren bij een DNS-request, kan worden geverifieerd dat het opgevraagde DNS-record afkomstig is van de juiste nameserver en niet is gewijzigd door een partij met kwade intenties. Wat DNSSEC **niet** doet, is het versleutelen van de inhoud van het verkeer, er kan enkel mee geverifieerd worden of het teruggegeven record authentiek is.

In de tab *DNSSEC signing* klik ik op *Enable DNSSEC signing*.

A screenshot of a computer

AI-generated content may be incorrect.

Ik selecteer de optie om een nieuwe Customer Managed Key aan te maken en voer de naam in in de *Provide KSK name* en *Create customer managed CMK* velden. Dan klik ik op *Create KSK and enable signing*. DNSSEC wordt nu ingeschakeld.



Na enkele momenten zien je dat het signen van DNSSEC succesvol is verlopen. Nu moet een chain of trust nog tot stand gezet worden. Ik klik op *View information to create DS record*.

A screenshot of a chat

AI-generated content may be incorrect.

Ik klap de velden *Establish a chain of trust* en *Route 53 registrar* open en kopieer de public key.

A screenshot of a computer

AI-generated content may be incorrect.

Nu navigeer ik naar *Route 53 -> Registered domains -> tristantech.org* en klik ik op de tab *DNSSEC keys*.

A screenshot of a computer

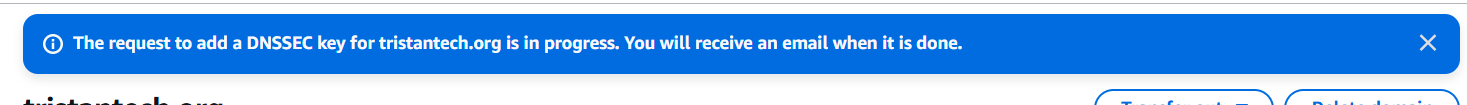
AI-generated content may be incorrect.

Ik klik op *Add key*, plak de Public key in het veld en klik weer op *Add*. De Key type en het algoritme hoeven niet aangepast te worden.

A screenshot of a computer

AI-generated content may be incorrect.

Een melding verschijnt dat de aanvraag om en DNSSEC key aan mijn domein toe te voegen is gedaan.



Enkele moment later ontvang ik een email.

A close-up of a computer screen

AI-generated content may be incorrect.

Nu is ook te zien dat de DNSSEC key is toegevoegd aan mijn domein.

A screenshot of a computer

AI-generated content may be incorrect.

Als ik nu naar <https://dnssec-debugger.verisignlabs.com/> navigeer en daar mijn domein invoer, zie ik dat het inrichten van DNSSEC succesvol is verlopen en er geen problemen zijn waargenomen.

A screenshot of a computer

AI-generated content may be incorrect.

**CloudFront**

**Om onze website werkend te krijgen moeten we de Origin Acces Control instellingen nog wijzigen.**

CloudFront distributie aanmaken met REST API endpoint.

Ik ga naar Amazon CloudFront en klik op *Create distribution*.

A screenshot of a computer

AI-generated content may be incorrect.

Bij het creëren van de distribution kies ik het bucket endpoint van mijn eerder aangemaakte bucket *tristantech.org*. Het is van belang om hier het bucket endpoint en niet het website endpoint te kiezen. Het bucket endpoint is namelijk een REST API endpoint en hiermee kunnen Origin access control setting worden ingericht. Op deze manier kan ik de bucket enkel publiek toegankelijk maken via mijn CloudFront distributie. Ik vink *Origin access control settings* dan ook aan en voeg hier mijn bucket aan toe.

A screenshot of a computer

AI-generated content may be incorrect.

Ik selecteer de optie *Redirect HTTP to HTTPS*.

A close-up of a computer

AI-generated content may be incorrect.

De *Web Application Firewall* zet ik aan.

A screenshot of a computer

AI-generated content may be incorrect.

Ik voeg twee *Alternate domain names* toe.

A screenshot of a computer

AI-generated content may be incorrect.

Ik voeg het SSL certificaat dat ik eerder heb aangemaakt in ACM toe zodat de verbinding wordt beveiligd met HTTPS.

A screenshot of a computer

AI-generated content may be incorrect.

En *Standard logging* schakel ik in.

A screenshot of a computer

AI-generated content may be incorrect.

Dan klik ik op *Create distribution.* De distribution is aangemaakt maar er verschijnt een melding dat de S3 bucket policy moet worden geupdate.

A screenshot of a computer

AI-generated content may be incorrect.

Ik kopieer de bucket policy en voeg hem toe aan mijn *tristantech.org* bucket.

A screenshot of a computer

AI-generated content may be incorrect.

Nu is het tijd om records toe te voegen aan onze hosted zone, zodat we uiteindelijk de websitecontent uit de S3 bucket kunnen ophalen.