

Project (Cloud6.Sentia1)

Introductie:

In de afgelopen maanden heb je veel geleerd over de cloud en deze kennis toegepast op relatief kleine opdrachten. Dit gaat nu allemaal samenkomen in de volgende *grote* opdracht waarin je een bestaande architectuur verbetert en automatiseert.

Je zal de komende weken samen aan individuele oplossingen werken voor je eigen cloud provider waarbij je te maken krijgt met bedrijfsbeleid, eisen van eindgebruikers, eisen van collega's, en de tijdsdruk van een groot project in een beperkte tijd. Hierin zullen de verwachtingen op het niveau van een junior engineer zijn.

Wat zijn de verwachtingen in dit project?

- Met weinig begeleiding nieuwe kennis opdoen en toepassen
- Op een abstract niveau het project begrijpen
- Zelfstandig problemen op kunnen lossen
- Een discussie kunnen aangaan over de technische elementen
- Verbeteringen kunnen suggereren op code- en projectniveau

BELANGRIJK: Je zult tijdens het project nieuwe kennis opdoen. Dit is heel normaal in de IT wereld. Zelfs de meest senior engineers zullen nog altijd nieuwe dingen moeten leren voordat ze aan een nieuw project beginnen. Concepten als Infrastructure as Code ga je nu ook daadwerkelijk toepassen en daarvan is verwacht dat je tijd nodig hebt om de details ervan te leren. Onderzoek is geen verspilde tijd! Leren is geen verspilde tijd!

Hieronder zie je de huidige architectuur van onze applicatie. Jij bent verantwoordelijk voor het verhuizen van de huidige servers naar de cloud en voor het automatiseren van de deployment van deze infrastructuur. Dit moet gedaan worden in CDK (voor AWS) of Bicep (voor Azure). Voorlopig is het genoeg om simpele placeholder content te gebruiken voor de website (denk aan een simpele index.html in plaats van een hele website).

We zullen werken met sprints van 2 weken waar je aan het eind van de sprint je progressie presenteert aan de stakeholders. Je zal uiteindelijk het project met de deliverables individueel inleveren via GitHub.

Benodigdheden:

- Je GitHub repository
- Project Templates
- Je labomgeving

Opdracht:

Je hebt een opdracht gekregen om een bedrijf te helpen bij de transitie naar de cloud. Het bedrijf heeft zijn infrastructuur laten analyseren door een eerder team. Er is een diagram gemaakt naar aanleiding van de huidige situatie. Je kan deze diagrammen vinden in de [Bijlagen](#).

Jij zal de Infrastructure as Code app bouwen om dit ontwerp naar de cloud te brengen. Het is de bedoeling dat je voor deze app CDK van AWS of Bicep van Azure gebruikt. De volgende eisen zijn aangegeven als noodzakelijk:

- Alle VM disks moeten encrypted zijn.
- De webserver moet dagelijks gebackupt worden. De backups moeten 7 dagen behouden worden.
- De webserver moet op een geautomatiseerde manier geïnstalleerd worden.
- De admin/management server moet bereikbaar zijn met een publiek IP.
- De admin/management server moet alleen bereikbaar zijn van vertrouwde locaties (office/admin's thuis)
- De volgende IP ranges worden gebruikt: 10.10.10.0/24 & 10.20.20.0/24
- Alle subnets moeten beschermd worden door een firewall op subnet niveau.
- SSH of RDP verbindingen met de webserver mogen alleen tot stand komen vanuit de admin server.
- Wees niet bang om verbeteringen in de architectuur voor te stellen of te implementeren, maar maak wel harde keuzes, zodat je de deadline kan halen.

In de uitwerking van de CDK/Bicep-app zorg ervoor dat je klein begint en dat je incrementeel features toevoegt. Zorg ervoor dat je altijd een commit / branch heb waar je op terug kan vallen met een werkende versie van je applicatie. Je kan met Git Tag commits labels geven die makkelijk in GitHub terug te vinden zijn. Mocht je code compleet zijn met de bovengenoemde eisen, dan kan je de tag 'v1.0' gebruiken.

Hoe gaan we te werk?

Het project kent twee fasen. De eerste fase duurt 5 weken waarin je zal werken aan v1.0 van de applicatie terwijl je ook leert programmeren in Python.

De tweede fase van het project zal een verandering in je architectuur introduceren. Dit simuleert een situatie waarbij een klant nieuwe eisen introduceert in je project en zal later vrijgegeven worden.

Je werkt individueel aan je eigen uitwerking. Je zal wel in teams werken van 3. De volgende afspraken zijn van toepassing:

- Een sprint is 2 weken, met uitzondering de eerste sprint die 3 weken is.
- Elke laatste vrijdag is er een sprint retrospective.
 - Hierin bespreekt het team het proces van de afgelopen week en daarin verbeteringen.
 - Uit deze verbeteringen zullen de actiepunten genoteerd worden in het retro-document. Deze actiepunten zijn nieuwe user stories voor de volgende sprint.
- Elke laatste vrijdag is er een sprint demo waarin de teams aan de product owners hun voortgang presenteren. Een aangeven hoe/waarom sprint behaald of niet behaald is.
- Op de laatste vrijdag van het project presenteert men het gehele project aan elkaar. Hierin deel je je ervaring, je design decisions, en de architectuur van je project.
- Elke team heeft één scrummaster.
 - De scrummaster gekozen in de eerste week draagt zijn/haar rol voor het gehele project.
 - De scrummaster is het eerste aanspreekpunt voor het team. Teamleden van verschillende teams moeten niet elkaar opzoeken voor hulp. Dit verstoort de focus.
 - Elke werkdag is op 14:00 een scrummaster meeting van max 15 minuten. De scrum masters overleggen hierin de voortgang van de teams. Hulp kan hier gevraagd en aangeboden worden.
 - De scrummaster is verantwoordelijk voor het trekken aan de bel wanneer het duidelijk wordt dat een sprint niet gehaald gaat worden. Dit doe je bij de product owners.
- De product owners zijn de learning coaches.

Voor dit project zullen verschillende scrum teams worden gevormd. Elk team zal dezelfde user stories uitwerken. Als je ergens niet uitkomt binnen jouw team betekent het dat je op andere teams kan terugvallen. Hoewel het begrijpelijk is dat elk team focus wil hebben op zijn eigen werk is het ook belangrijk om in de gaten te houden wat de rest aan het doen is. Kennisdeling is essentieel van belang om voortgang te maken in jouw werk proces. Het is de taak van de scrum master om overzicht te bewaken over wat er bij andere teams gebeurt. Gebruik elkaars kennis dus ook om de voortgang van het project te versoepelen/versnellen.

Wat kan men qua ondersteuning verwachten van andere teams:

- Resources beschikbaar stellen om jouw team op weg te helpen (video's, artikelen, boeken, internet etc)
- Suggesties geven op wat je eventueel kan doen om jouw probleem op te lossen.

Hiermee zou het voldoende moeten zijn. Op dit punt zou jij weer verder kunnen gaan met de volgende stap. Ondersteuning van andere teams is dus jouw op weg helpen en niet jouw probleem oplossen.

Wat kan men qua ondersteuning **niet** verwachten van andere teams:

- Een uur of langer coding session doen om jouw probleem te tackelen.
- Reeks aan vragen afvuren en verwachten dat een ander team daar alle antwoorden op kan geven.
- Ondersteuning krijgen op vragen of problemen die niet helder zijn. Als een vraag voor de ander niet duidelijk is, dan mag zij/hij afwijzen.

Wat kan men verwachten van de learning coaches:

- De learning coaches zijn de product owners van het project
- De learning coaches zijn aanwezig tijdens de standaard contactmomenten (dagopening, Q&A, dagafsluiting)
- Voor andere contactmomenten moeten afspraken gemaakt worden vanuit de scrum masters naar de learning coaches
- Zorg ervoor dat je je vragen duidelijk en helder geformuleerd hebt.

Definition of Done

Wanneer is een user story af?

- Als de user story een deliverable heeft: wanneer deze op Github staat.
- Als de user story over een stuk IaC gaat: wanneer deze geïsoleerd gedeployed kan worden zonder foutmeldingen.
- Wanneer de documentatie over de user story op Github bijgewerkt is.

Wanneer is een versie van de app af?

- v1.0: Als deze voldoet aan alle eisen zoals aangegeven in dit document
- v1.1: Als deze voldoet aan alle nieuwe eisen.

Deliverables:

De volgende deliverables worden verwacht in je GitHub repository aan het eind van dit project:

- Een werkende CDK / Bicep app van het MVP
- Ontwerp Documentatie
- Beslissing Documentatie
- Tijd logs
- Eindpresentatie

Werkende applicatie van het MVP

De werkende applicatie moet een build en een deployment succesvol afronden. Een versie van je MVP moet eenvoudig te identificeren zijn. Dit kan met een tag, of een release.

Daarnaast moet je repository documentatie bevatten over hoe je de applicatie gebruikt. Hierin geef je aan hoe je de applicatie aanroept, de argumenten die nodig zijn, en welke rechten deze nodig heeft in AWS of Azure om te deployen.

Ontwerp Documentatie

Je zal de bestaande architectuur gebruiken. Er zijn nog wel details die verder uitgewerkt dienen te worden. In dit document zal je de gaten opvullen en uiteindelijk de praktische en technische informatie vermelden in GitHub. Dit document zal informatie bevatten over je gekozen (N)SG regels. Maar ook een visualisatie van wat, en in welke volgorde, je applicatie deployt in de cloud.

In v1.1 zal je hier ook je eigen diagrammen plaatsen voor de aanpassingen en de verbeteringen onderbouwen.

Beslissing Documentatie

Tijdens het implementeren van het ontwerp zal je beslissingen maken over o.a. diensten die je gaat gebruiken. In dit document zal je je overwegingen uitschrijven en je besluiten onderbouwen. Dit document zal ook al je assumpties en verbeteringen bevatten. Dit dient als basis voor je ontwerp documentatie.

Tijd logs

Dit is bestand waar je in gestructureerd je dagen bijhoudt. Je geeft aan in 1 enkele zin waar je aan gewerkt hebt die dag. Puntsgewijs zal je de obstakels aangeven die je hebt ervaren en de oplossingen die je hebt gevonden.

Je kan de volgende template gebruiken:

```
# Log [datum]

## Dagverslag (1 zin)

## Obstakels

## Oplossingen

## Learnings

---
```

Tussentijdse Presentaties

Je zal aan het eind van iedere sprint een presentatie maken van je voortgang. Hierin behandel je wat je de afgelopen sprint heb gedaan. En bij de oplevering laat je een demo zien van je werk.

Eindpresentatie

Op de laatste vrijdag van het project presenteert men het gehele project aan elkaar. Hierin deel je je ervaring, je design decisions, en de architectuur van je project.

User Stories

De Product Owners hebben al een overleg gehad en hebben de volgende epics en backlog opgesteld. Jij gaat als team aan deze epics en user stories werken.

Mocht je met je team user stories identificeren dat opgesplitst moet worden in kleinere stories, is je team hier vrij in.

De volgende epics zijn van toepassing voor het project:

Exploratie	De exploratie epic is bedoeld om keuzes te maken over het project. Als je deze epic hebt afgesloten, zorg ervoor dat je er niet op terug hoeft te komen. Je hebt te weinig tijd om grote aanpassingen door te voeren.
v1.0	Epic v1.0 is het opleveren van de Infrastructure as Code en alle vereiste documentatie naar de eisen die je hebt ontdekt tijdens de exploratie.
v1.1	Versie 1.1 is het opleveren van de Infrastructure as Code en alle vereiste documentatie naar de eisen die later beschikbaar worden.

Hier volgen de user stories voor de epics exploratie, en v1.0:

Als team willen wij duidelijk hebben wat de eisen zijn van de applicaties	
Epic	Exploratie
Beschrijving	Je hebt al heel wat informatie gekregen. Er staan al wat eisen in dit document vernoemt. Alleen deze lijst is mogelijk incompleet of onduidelijk. Het is belangrijk om alle onduidelijkheden uitgezocht te hebben voordat je groot werk gaat verzetten.

Als team willen wij duidelijk hebben wat de eisen zijn van de applicaties

Deliverable	Een puntsgewijze omschrijving van alle eisen
-------------	--

Als team willen wij een duidelijk overzicht van de aannames die wij gemaakt hebben.

Epic	Exploratie
Beschrijving	Je hebt al heel wat informatie gekregen. Mogelijk zijn er vragen die geen van de stakeholders heeft kunnen beantwoorden. Je team moet een overzicht kunnen produceren van de aannames die je daardoor maakt.
Deliverable	Een puntsgewijze overzicht van alle aannames

Als team willen wij een duidelijk overzicht hebben van de Cloud Infrastructuur die de applicatie nodig heeft

Epic	Exploratie
Beschrijving	Je hebt al heel wat informatie gekregen. En al een ontwerp. Alleen in het ontwerp ontbreken nog zaken als IAM/AD. Identificeer deze extra diensten die je nodig zal hebben en maak een overzicht van alle diensten.
Deliverable	Een overzicht van alle diensten die gebruikt gaan worden.

Als klant wil ik een werkende applicatie hebben waarmee ik een veilige netwerk kan deployen

Epic	v1.0
Beschrijving	De applicatie moet een netwerk opbouwen dat aan alle eisen voldoet. Een voorbeeld van een genoemde eis is dat alleen verkeer van trusted sources de management server mag benaderen.
Deliverable	IaC-code voor het netwerk en alle onderdelen

Als klant wil ik een werkende applicatie hebben waarmee ik een werkende webserver kan deployen

Epic	v1.0
Beschrijving	De applicatie moet een webserver starten en deze beschikbaar maken voor algemeen publiek.
Deliverable	IaC-code voor een webserver en alle benodigdheden

Als klant wil ik een werkende applicatie hebben waarmee ik een werkende management server kan deployen

Epic	v1.0
Beschrijving	De applicatie moet een management server starten en deze beschikbaar maken voor een beperkt publiek.
Deliverable	IaC-code voor een management server met alle benodigdheden

Als klant wil ik een opslagoplossing hebben waarin bootstrap/post-deployment script opgeslagen kunnen worden

Epic	v1.0
Beschrijving	Er moet een locatie beschikbaar zijn waar bootstrap scripts beschikbaar worden. Deze script moeten niet publiekelijk toegankelijk zijn.
Deliverable	IaC-code voor een opslagoplossing voor scripts

Als klant wil ik dat al mijn data in de infrastructuur is versleuteld

Epic	v1.0
Beschrijving	Er wordt veel gehecht aan de veiligheid van de data at rest en in motion. Alle data moet versleuteld zijn met een encryptie sleutel in het beheer van de klant.
Deliverable	IaC-code voor versleuteling voorzieningen

Als klant wil ik iedere dag een backup hebben dat 7 dagen behouden wordt

Epic	v1.0
Beschrijving	De klant wil graag dat er een backup beschikbaar is, mocht het nodig zijn om de servers terug te brengen naar een eerdere staat. (Zorg ervoor dat de Backup ook daadwerkelijk werkt)
Deliverable	laC-code voor backup voorzieningen

Als klant wil ik weten hoe ik de applicatie kan gebruiken

Epic	v1.0
Beschrijving	Zorg dat de klant kan begrijpen hoe deze de applicatie kan gebruiken. Zorg dat het duidelijk is wat de klant moet configureren voor de deployment kan starten en welke argumenten het programma nodig heeft.
Deliverable	Documentatie voor het gebruik van de applicatie

Als klant wil ik een MVP kunnen deployen om te testen

Epic	v1.0
Beschrijving	De klant wilt zelf intern je architectuur testen voordat ze de code gaan gebruiken in productie. Zorg ervoor dat er configuratie beschikbaar is waarmee de klant een MVP kan deployen.
Deliverable	Configuratie voor een MVP deployment

Belangrijke data:

Hier volgt een overzicht van belangrijke mijlstenen in het project:

Onderwerp:	Datum (projectweek):
Start Python, Start Project (v1.0)	07-02-2022 (wk 1)
Introductie Project v1.1	14-03-2022 (wk 5)
Oplevering- / Eindpresentatie	08-04-2022 (wk 9)

Hou ook rekening met de volgende projectactiviteiten:

Project Activiteit:	Datum (projectweek) :
Sprint 1 Review progressie app v1.0	25-02-2022 (wk 3)
Sprint 2 Review oplevering app v1.0	11-03-2022 (wk 5)
Sprint 3 Review progressie app v1.1	25-03-2022 (wk 7)
Sprint 4 Review oplevering app v1.1 / Eindpresentatie	08-04-2022 (wk 9)

Resources:

Voor het ontwerpen van je architectuur: [Draw.io](https://draw.io)

AWS CDK documentatie: [link](#)

Cloudformation documentatie: [link](#)

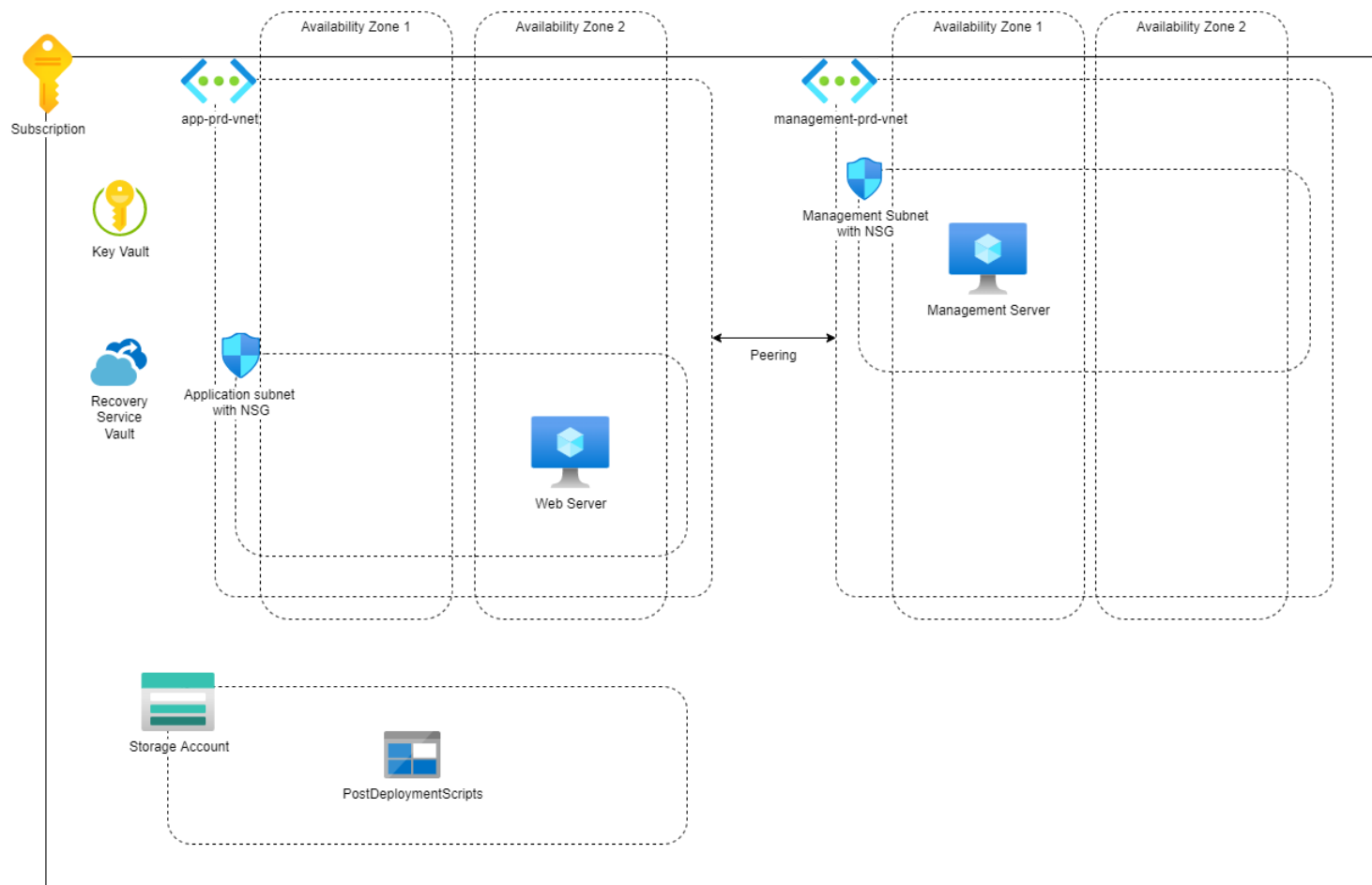
Cloudformation resource omschrijvingen: [link](#)

Azure Bicep documentatie: [link](#)

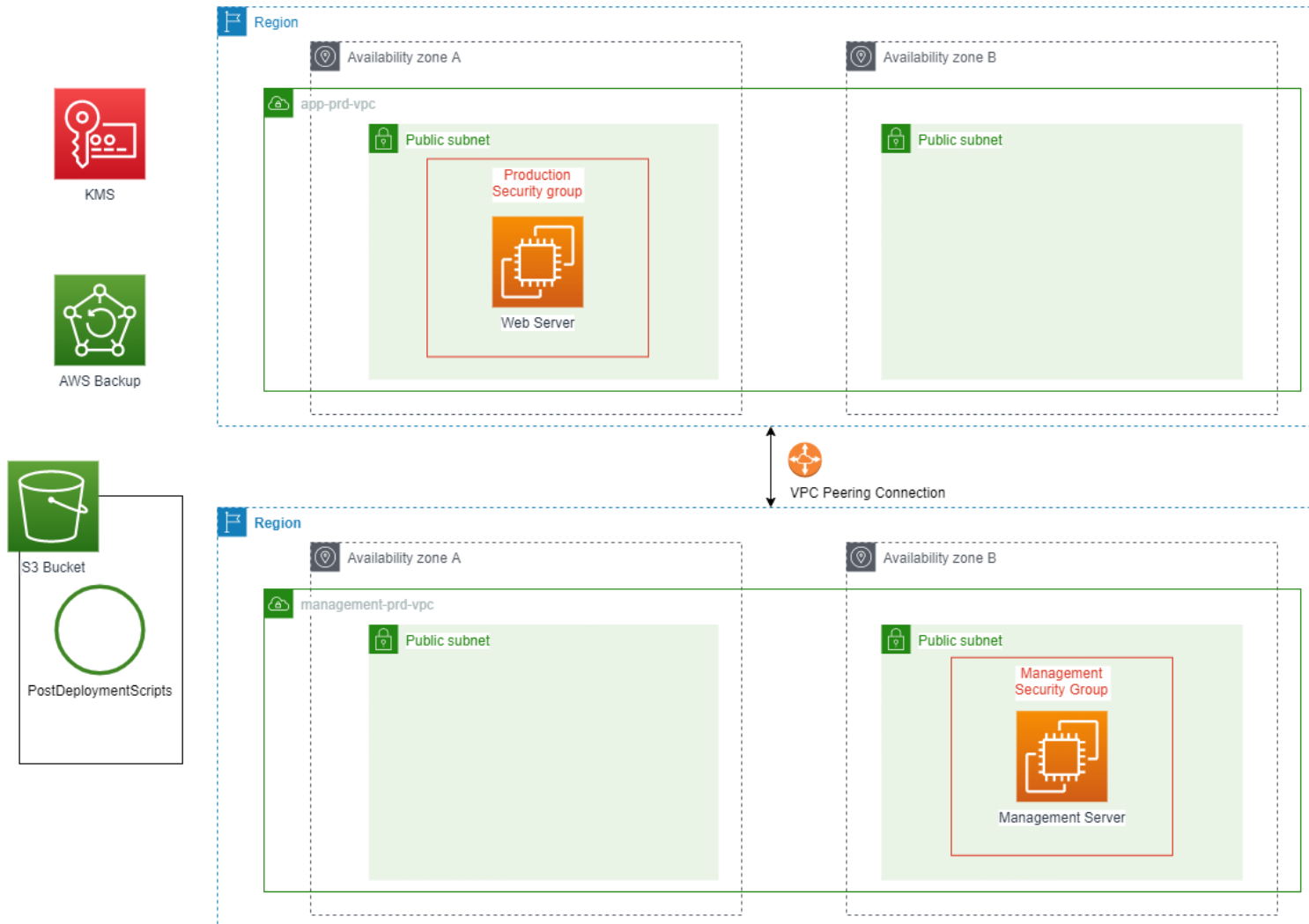
Azure ARM template documentatie: [link](#)

Azure ARM resource omschrijvingen: [link](#)

Bijlagen



Figuur 1: Ontwerp in Azure



Figuur 2: Ontwerp in AWS