Stappenplan Cloud-systemen.

# Deel 1

Voor deel 1 moesten we een VM aanmaken en de website via deze VM online zetten.

Dit is hoe ik mijn site heb online gezet.

Eerst en vooral surfen we naar portal.azure.com

Daar selecteren we “education”

Dan gaan we naar Gratis services: alles verkennen.

Hier kiezen we de LINUX VM er is er ook eentje van windows maar die mag je negeren.

Bij de linux machine vullen we alle informatie in, genereren we een ssh key die we opslagen op een veilige plaats en ga ik de poort instellen op ssh 20.

Daarna duwen we op maken

Afbeelding met tekst, schermopname, software, Computerpictogram

Automatisch gegenereerde beschrijving

Hierna gaan we naar de overzichtspagina door op naar resource gaan te duwen.

Afbeelding met tekst, schermopname, software, Webpagina

Automatisch gegenereerde beschrijving

Vervolgens loggen we in met eigen ssh. De 2de optie.

Afbeelding met tekst, software, Computerpictogram, Webpagina

Automatisch gegenereerde beschrijving

Hiervoor hebben we onze SSH sleutel en de terminal nodig.

Dus open je terminal en vul de link in die je hebt gekregen bij inloggen met eigen SSH. Afbeelding met tekst, software, Computerpictogram, computer

Automatisch gegenereerde beschrijving

Vervolgens moeten we hier nano installeren voor enkele scripts te maken.

Dit doe je door sudo apt-install nano in te geven.

Eens we een script hebben opgesteld met daarin de werking voor de apacha web server, kunnen we dit opslaan, de poort voor http traffic instellen op 80 en zo kunnen we vervolgens een script aanmaken voor de databank te koppelen

Afbeelding met tekst, software, computer, schermopname

Automatisch gegenereerde beschrijving

Afbeelding met tekst, software, Computerpictogram, Webpagina

Automatisch gegenereerde beschrijving

Nadat we de config script gerund hebben kunnen we vervolgens eens de mappenstructuur bekijken die we nu hebben:

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

## FreeDNS:

1. Eerst gaan we naar “Freedns.afraid.org”
2. Hier maak je een gratis account aan en navigeer je naar subdomains in het aside-menu.
3. Hier kan je vervolgens een subdomein toevoegen (of indien je deze al hebt, aanpassen)

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Nadat je op “Add” hebt geklikt kan je voort met de volgende menu:

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

1. Het type voor het subdomain, dit houden we op A zo kunnen we connecteren met een IP adress.
2. Hier vul je in welke naam je subdomain moet hebben, bij mij is dit dus HeteSaus
3. Hier kies je een publiek domein, je hebt er velen maar ik kies standaard voor chickenkiller.com (dit is een public domain)
4. Hier vul je het IP adress in naar waar je gerouteerd moet worden nadat er dus naar HeteSaus.chickenkiller.com word gesurft, dit is het publieke IP-adress van onze VM.
5. 7 moet eigenlijk 5 zijn: Hier vul je de security code in. (captcha)
6. Als we dan op save duwen wordt je domein opgezet.

Bij mij is het dus geworden: <http://hetesaus.chickenkiller.com>

Indien we hierheen surfen krijgen we dus onze website te zien:

Afbeelding met tekst, drinken, schermopname, software

Automatisch gegenereerde beschrijving

Als dit klopt gaan we door met het databank gedeelte.

Dit is wat de console zoet moeten teruggeven wanneer je wilt connecten met de databank.Afbeelding met tekst, elektronica, schermopname, software

Automatisch gegenereerde beschrijving

Met een 2de terminal ziet dit er zo uit:

Afbeelding met tekst, elektronica, schermopname, software

Automatisch gegenereerde beschrijving

Vervolgens moeten we elke keer er iemand op deze site komt dit loggen in een databank.

Dit werk op de VM momenteel foutloos, en ik run het local met de MYSQL: deze zal ik even tonen: (dit is een screenshot van deel 2 dus de databank en tabel hebben een andere naamgeving)

Voor de database na te kijken op de VM gebruik je: docker exec -it “de naam van je container” mariadb -u root -p

# 

**Ik maak een filmpje met alle delen tesamen. Dit houdt het overzichtelijker.**

# Deel 2

Voor deel 2 moesten we een dockercompose en Dockerfile aanmaken voor de automatische opstart van de containers en de website.

Eerst maak ik mijn docker compose file:

Afbeelding met tekst, schermopname, software, Computerpictogram

Automatisch gegenereerde beschrijving

Dan creeer ik een dockerFile:

Afbeelding met tekst, nummer, Lettertype, software

Automatisch gegenereerde beschrijving

vervolgens kan je naar je VM gaan en daar:

1. **git installeren:**

**Afbeelding met tekst, Lettertype, schermopname, nummer

Automatisch gegenereerde beschrijving**

1. **docker installeren en mysql\_secure installeren:**

**Afbeelding met tekst, schermopname, Lettertype, nummer

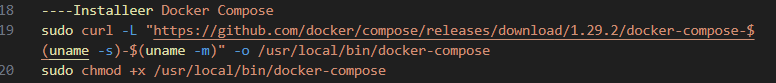
Automatisch gegenereerde beschrijving**

1. **Opnieuw inloggen om de veranderingen door te voeren.**

**Afbeelding met tekst, Lettertype, schermopname, lijn

Automatisch gegenereerde beschrijving**

1. **docker-compose installeren:**

****

Dan staat alles klaar op de VM voor de opzet van de containers dus dan gaan we als volgt:

1. **Clone je project naar een map op de VM met:**

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

1. **De containers opstarten met het UP commando:**

Afbeelding met tekst, schermopname, Lettertype, software

Automatisch gegenereerde beschrijving

1. **Nakijken of de container actief is met het commando docker ps:**

**Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving**

1. **Dan kan je naar de site surfen door naar het IP van je VM te gaan gevolgd door 8080**

Afbeelding met tekst, drinken, schermopname, software

Automatisch gegenereerde beschrijving

Inloggen in de databank:

1. **Connect in een nieuwe terminal naar je VM en navigeer naar de juiste map of in de huidige terminal**
2. **Connect dan met het volgende commando: docker exec -it “Je container-Naam” mariadb -u root -p**
3. **Vul je wachtwoord in, bij mij is dit rootWord**
4. **Navigeer naar de juiste table en vraag de data op met :**

**USE visitors;**

**SELECT \* from visit\_logs;**

**Afbeelding met tekst, schermopname, Lettertype, software

Automatisch gegenereerde beschrijving**

**Na een korte refresh en nadat iemand anders er naartoe gesurft is doe ik dit nog eens:**

Afbeelding met tekst, schermopname, Lettertype, zwart-wit

Automatisch gegenereerde beschrijving

Samenvatting deel 2:

**We hebben voor deel 2 gemaakt:**

1. Een docker-compose file: Deze hebben we nodig om mee te geven hoe bepaalde containers moeten worden opgestart
2. Een dockerFile
3. Ervoor gezorgd dat deze vlekkeloos de containers opstarten zodat je met een simpel commando de website online kunt zetten : docker-compose up -d

**Ik maak zoals eerder vermeld een video voor alle delen tesamen.**

# Deel 3: Github workflow.

Voor deel 3 was het de bedoeling dat we de een github workflow toevoegen aan ons project:

Elke keer er een aanpassing word gepushed in main moet deze een nieuwe image bouwen.

Hiervoor hebben we een CI.yml nodig: hierin zet ik mijn code voor de workflow

En ook een paar github secrets: dit leg ik zo uit.

1. **Eerst maken we de secrets aan:**

**Afbeelding met tekst, schermopname, software, Computerpictogram

Automatisch gegenereerde beschrijving**

**1) Eerst navigeer je naar je juiste repository.**

**2) Vervolgens duw je op settings**

**3) Dan gaan we naar secret en variables**

**4) Dan naar actions**

**5) Daar duwen we dan op new repository secret**

**6) Hier vullen we in DOCKER\_USERNAME (hier zetten we dan de naam van op docker hub) en DOCKER\_PASSWORD (dit is het password van dockerhub)**

1. **Dan maken we de workflow code aan: Ik zet hier mijn commentaar erbij in.**

**Afbeelding met tekst, software, schermopname

Automatisch gegenereerde beschrijving**

1. **We passen de docker-compose file vervolgens een beetje aan:**

**Afbeelding met tekst, schermopname, software

Automatisch gegenereerde beschrijving**

**We moeten nu een image gebruiken IPV een build. We geven als image onze eigen image mee met de latest tag.**

1. **Vervolgens zorgen we ervoor dat dit in de juiste structuur in de map staat**

**Afbeelding met tekst, schermopname, software, Computerpictogram

Automatisch gegenereerde beschrijving**

1. **Dan kijk ik na of ik de repository voor de image heb aangemaakt op docker hub:**

**Afbeelding met tekst, schermopname, software, Computerpictogram

Automatisch gegenereerde beschrijving**

1. **Dan doen we eens een push om te kijken of deze action succesvol uitvoert:**

**Dus ik maak een kleine aanpassing in mijn repository en push deze dan door:**

**(Ik heb de index-links aangepast in mijn html)**

**Afbeelding met tekst, schermopname, software, nummer

Automatisch gegenereerde beschrijving**

**Zoals je hierboven kunt zien kan je op details duwen om de werking te volgen.**

**Afbeelding met schermopname, tekst, software, Multimediasoftware

Automatisch gegenereerde beschrijvingZoals u kunt zien is de job succesvol afgesloten. Dat betekent dat nu de nieuwste image online staat op docker hub, we zullen dit nog eens even dubbelchecken:**

Afbeelding met tekst, schermopname, software, Computerpictogram

Automatisch gegenereerde beschrijving

**En zoals u ziet is dit inderdaad het geval. De image staat online met de meest recente build. Dus nu bouwen we onze container altijd met de recenste image.**

Samenvatting:

**We hebben een workflow toegevoegd die wanneer we een wijziging maken en pushen in onze main branch het volgende zal doen:**

1. **De broncode van de repository klonen.**
2. **Een Docker-image bouwen met de code in de repository.**
3. **Inloggen op Docker Hub met github en docker secrets.**
4. **Het Docker-image naar Docker Hub pushen met de tag “:latest” zodat deze klaarstaat voor volgend gebruik.**
5. **Dit zorgt ervoor dat er altijd een up-to-date Docker-image beschikbaar is op Docker Hub elke keer dat er veranderingen in de main branch worden doorgevoerd.**

## Hier vind u de link naar mijn github repository waar alle code staat:

<https://github.com/LarsLauryssensStudent/Project-Cloud-LarsL>

## De video vind u bijgeleverd bij de inzending en via deze link:

<https://ap.cloud.panopto.eu/Panopto/Pages/Sessions/List.aspx?folderID=e8f135ef-1858-44e8-bd3d-b18100ea0e9e>

Dit was mijn stappenplan, hopelijk voldoet het aan alle wensen, Lars Lauryssens