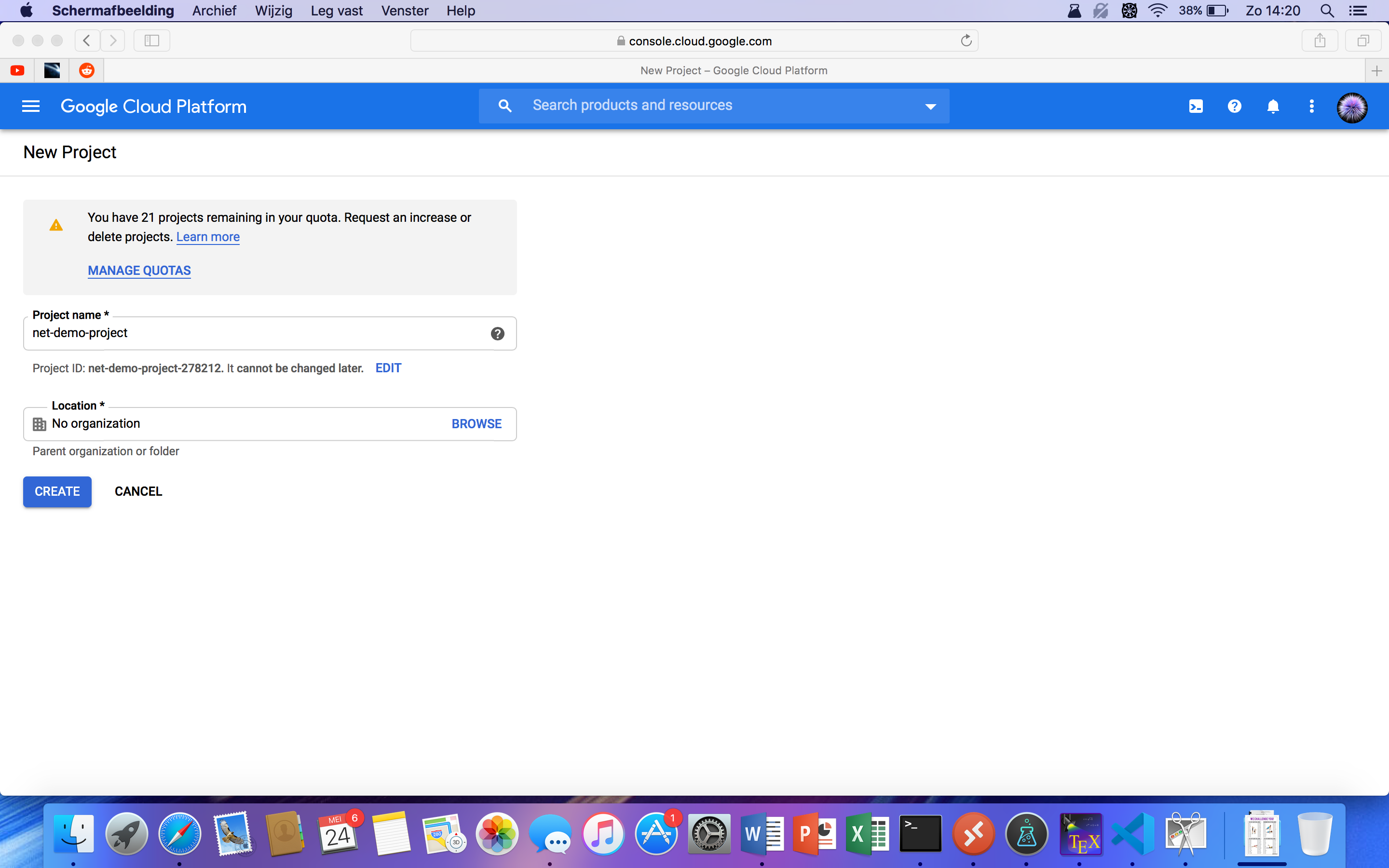
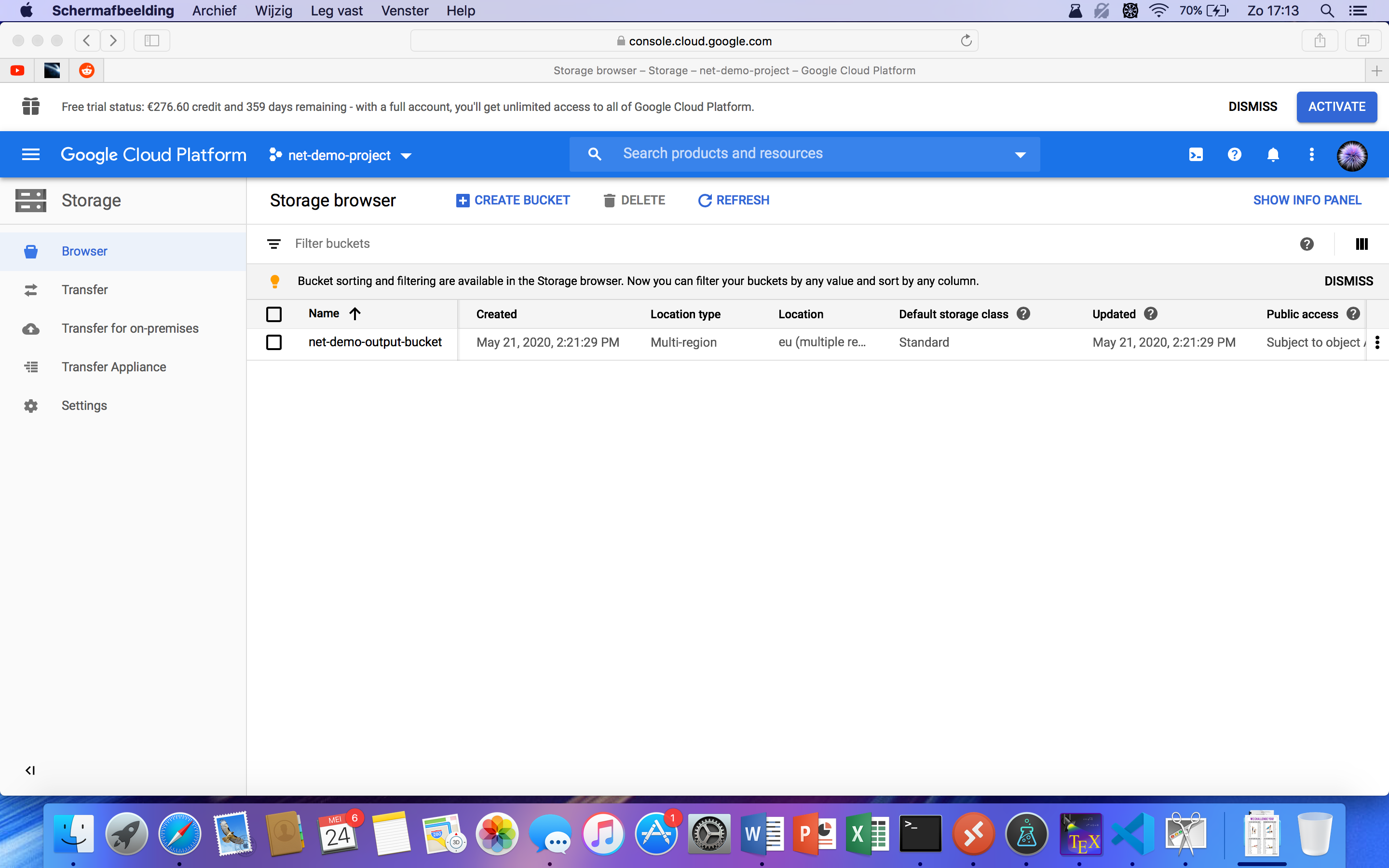
*Op de console van Google Cloud is er geen mogelijkheid om manueel een nieuwe pijpleiding aan te maken. De pijpleiding en de verschillende logs kunnen wel bekeken worden op het portaal. Om een pijpleiding aan te maken moet eerst de Google Cloud CLI (verwijzing) geïnstalleerd worden. Deze CLI is beschikbaar op ieder besturingssysteem en staat de gebruiker toe om alle services en producten van GCP aan te sturen en te beheren. De installatie ervan is simpel. Zeker met behulp van de documentatie van Google zelf. Documentatie link (verwijzing). Ook moet er een recente versie van Python beschikbaar zijn op de gebruiker zijn computer. Na deze geïnstalleerd te hebben moet het Google Cloud account geconnecteerd worden aan de CLI. Deze documentatie legt dit zorgvuldig uit. (Verwijzing).*

Voor deze POC is er een nieuw project aangemaakt op Google Cloud. Als op Google Cloud een project verwijderd wordt, worden ook alle resources en aanrekeningen stopgezet. Ook omdat er dan per project producten en services toegewezen kunnen worden. Zo kan er optimaal voldaan worden aan de gebruiker zijn noden. Het nieuw gecreëerde project, heeft de naam ‘net-demo-project’ gekregen zoals in figuur (verwijzing).



Om Code Build op Google Cloud te gebruiken moest eerst de API via de online console geactiveerd worden. Ook moest er nog een Storage Bucket aangemaakt worden voor het opslaan van de gecompileerde applicatie. De Storage service is standaard geactiveerd en hoefde dus niet aangezet te worden. Deze Storage Bucket heeft de naam ‘net-demo-output-bucket’ gekregen. Ook belangrijk was de selectie voor de locatie van deze Storage Bucket. Hier is er gekozen voor een geografisch zo dicht mogelijke locatie. Dit om de overdracht tijden van bestanden zo minimaal mogelijk te houden. Alle andere opties zijn onveranderd gebleven. Ook is er de mogelijkheid om toegangsrechten toe te kennen. Dit kan interessant zijn voor een productie omgeving. Figuur (verwijzing) toont de gecreëerde Storage Bucket en de geselecteerde opties.



Hierna is er een nieuwe map aangemaakt op de gebruiker zijn computer voor de applicatie. Deze is geïnitialiseerd als een Git repositorie. Hierin zijn dan de bestanden (verwijzing) voor de .Net applicatie in geplaatst. Ook het script (verwijzing) voor de bestanden overdracht met SFTP is hierbij toegevoegd. Hiernaast is er ook een gitignore aangemaakt. Figuur (verwijzing) toont een tree van de bestanden en mappen structuur.

poc\_gcp\_dotnet/

├── .git

├── .gitignore

├── MessageUtil

│   ├── MessageUtil.csproj

│   └── MessageUtilProgram.cs

├── MessageUtil.sln

├── MessageUtilTest

│   ├── MessageUtilTest.csproj

│   └── MessageUtilTests.cs

└── filetrans.sh

Vervolgens moest er connectie gemaakt worden met het gecreëerde project op Google Cloud met Google Cloud CLI. Hiervoor is op de CLI van de gebruiker zijn computer ‘gcloud init –console only’ uitgevoerd. Vervolgens is de wizard gevolgd en het correcte project geselecteerd. Deze figuur (verwijzing) toont een voorbeeld.

MBP-van-Kenzie:poc\_gcp\_dotnet kenzie$ gcloud init --console-only

Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [default] are:

core:

account: kenziec46@gmail.com

disable\_usage\_reporting: 'True'

project: net-demo-project

Pick configuration to use:

[1] Re-initialize this configuration [default] with new settings

[2] Create a new configuration

Please enter your numeric choice: 1

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:

gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.

Checking network connection...done.

Reachability Check passed.

Network diagnostic passed (1/1 checks passed).

Choose the account you would like to use to perform operations for

this configuration:

[1] kenziec46@gmail.com

[2] Log in with a new account

Please enter your numeric choice: 1

You are logged in as: [kenziec46@gmail.com].

Pick cloud project to use:

[1] net-demo-project

[2] tracx-1546698150141

[3] Create a new project

Please enter numeric choice or text value (must exactly match list

item): 1

Your current project has been set to: [net-demo-project].

Not setting default zone/region (this feature makes it easier to use

[gcloud compute] by setting an appropriate default value for the

--zone and --region flag).

See https://cloud.google.com/compute/docs/gcloud-compute section on how to set

default compute region and zone manually. If you would like [gcloud init] to be

able to do this for you the next time you run it, make sure the

Compute Engine API is enabled for your project on the

https://console.developers.google.com/apis page.

Your Google Cloud SDK is configured and ready to use!

Als volgende moest de configuratie van de CI/CD pijpleiding gemaakt worden. Zoals voorheen aangehaald werkt Google Cloud Code Build met Docker containers om al de gewenste taken uit te voeren. Voor deze POC is er gekozen om een Docker container van Microsoft zelf te gebruiken. Er is gekozen voor een .Net Core SDK container op DockerHub (verwijzing). Deze container wordt onderhouden door Microsoft zelf en heeft daarnaast ook uitgebreide documentatie ter beschikking op GitHub (verwijzing). Zo moest er geen aangepaste container gemaakt worden die in de toekomst dan problemen zou kunnen krijgen naarmate de software update. Deze container is gebaseerd op een licht gewicht Linux machine. Hierop staat dan de SDK van Microsoft om .Net applicaties te testen, compileren, publiceren, enz.

Pijpleidingen voor CI op Google Cloud Code Build worden geconfigureerd met behulp van een YAML-bestand. Voor deze POC zijn er vier verschillende stappen gedefinieerd met elk hun eigen doel. De eerste drie stappen maken allemaal gebruik van de door Microsoft gemaakte Docker container, ‘mcr.microsoft.com/dotnet/ore/sdk:3.1’. De laatste machine is een Ubuntu container. In de eerste stap worden de testen in ‘MessageUtilTest’ (verwijzing) uitgevoerd. De tweede stap compileert de code van de applicatie specifiek voor een 64 Bit Windows 10 platform en maakt een folder publish aan. In een derde stap wordt het Bash script ‘tarring.sh’ uitgevoerd dat deze publish map comprimeert. Figuur (verwijzing) toont dit Bash script. In een laatste stap wordt het filetrans script van figuur (verwijzing) uitgevoerd. Ook is er een stukje code voorzien dat het gemaakte ‘artifact’ (de gecomprimeerde map) gaat uploaden naar een bucket op Google Cloud. Figuur (verwijzing) toont deze cloubuild.yaml. Google Cloud Code Build gaat automatisch bij de overgang van iedere stap naar een andere virtuele machine, het volume waarin gewerkt wordt mee monteren. Hierdoor zijn er geen speciale stappen of acties nodig om bestanden tussen de verschillende machines uit te wisselen.

steps:

- name: 'mcr.microsoft.com/dotnet/core/sdk:3.1'

entrypoint: 'dotnet'

args: [ 'test' ]

- name: 'mcr.microsoft.com/dotnet/core/sdk:3.1'

entrypoint: 'dotnet'

args: [ 'publish', '-c', 'Release', '-r', 'win10-x64' ]

- name: 'mcr.microsoft.com/dotnet/core/sdk:3.1'

entrypoint: 'bash'

args: [ './tarring.sh' ]

- name: 'ubuntu'

entrypoint: 'bash'

args: [ './filetrans.sh']

artifacts:

objects:

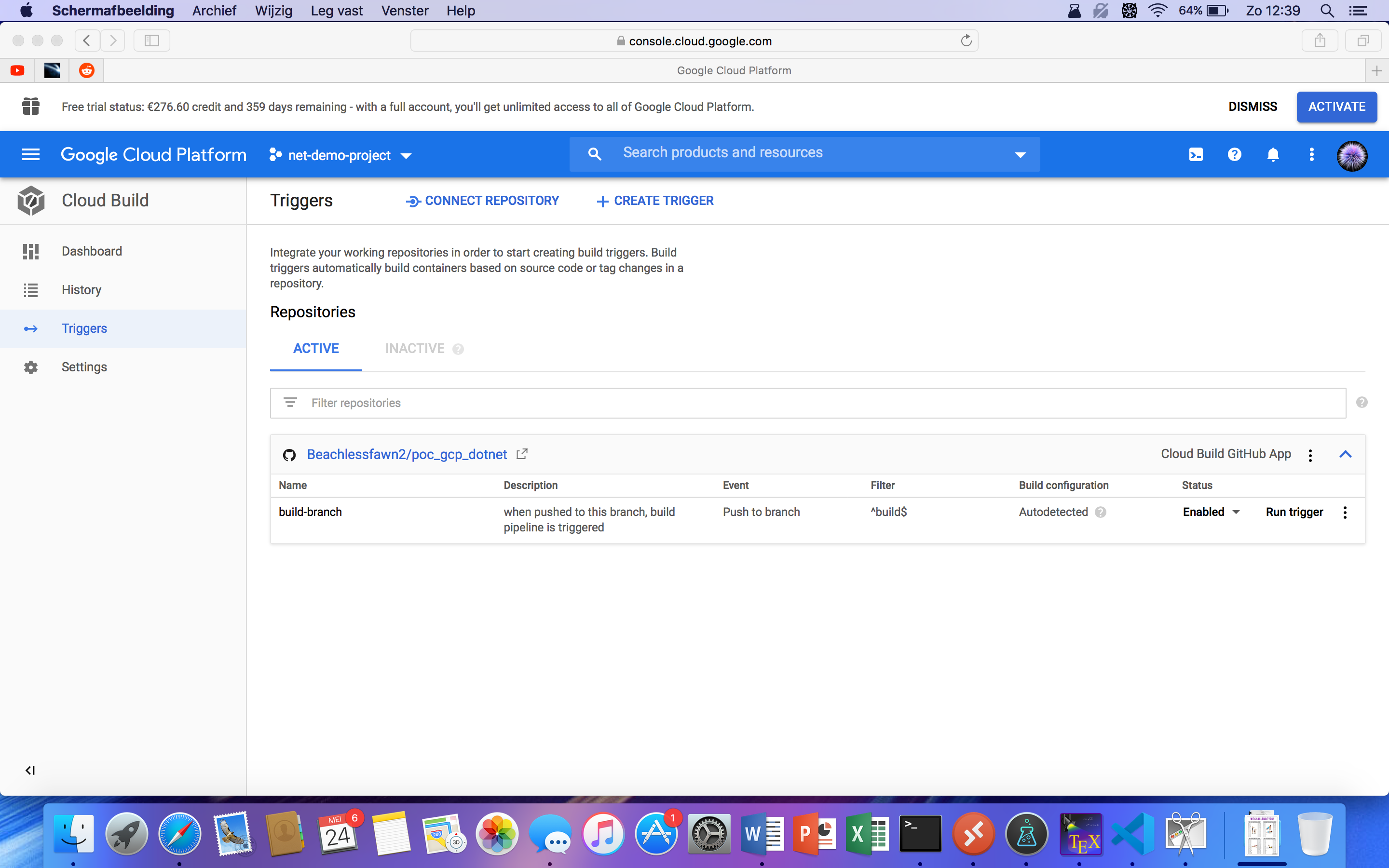
location: 'gs://net-demo-output-bucket/'

paths: ['/workspace/MessageUtil/bin/Release/netcoreapp3.1/win10-x64/messageutil-win10-x64.tar.gz']

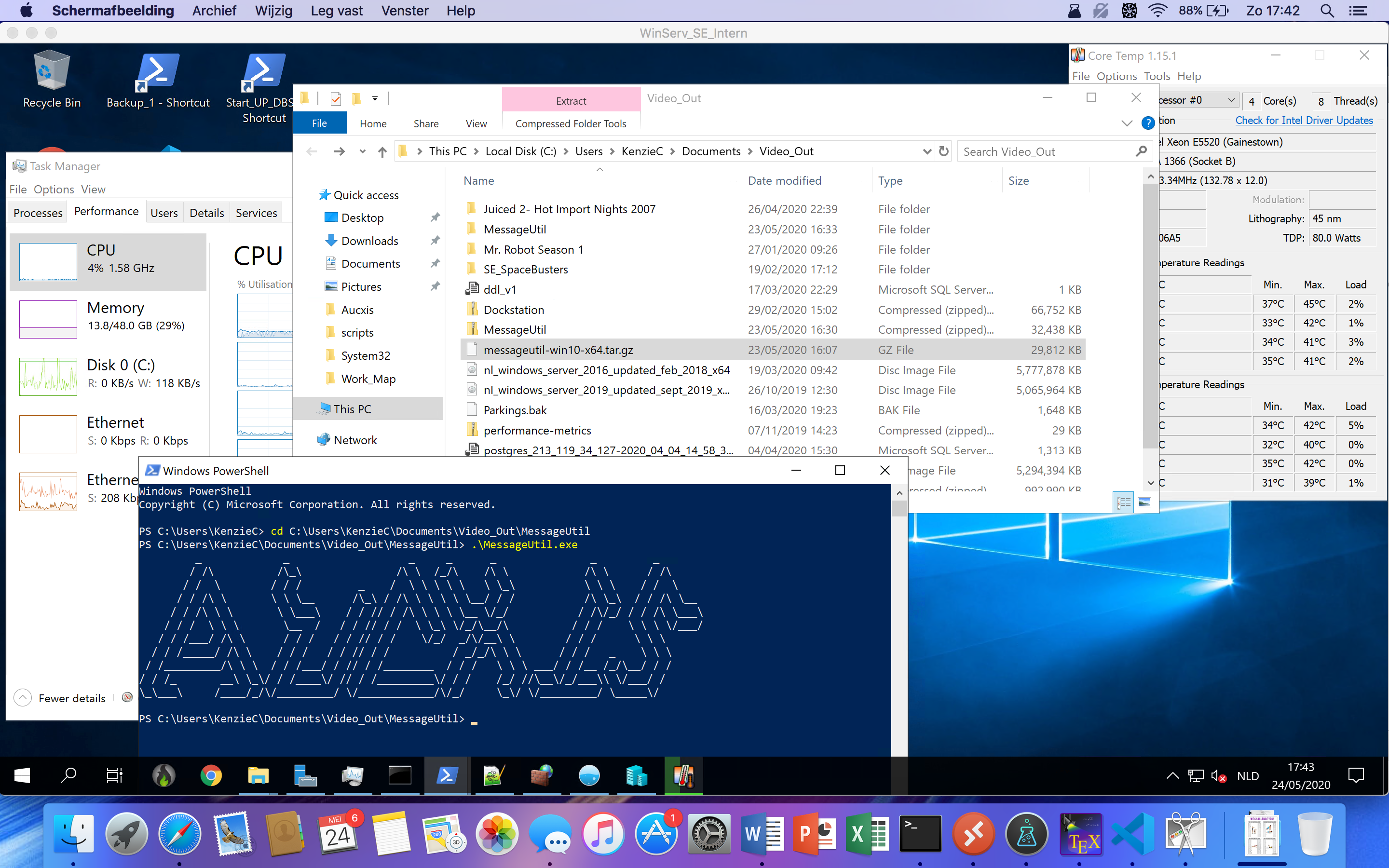
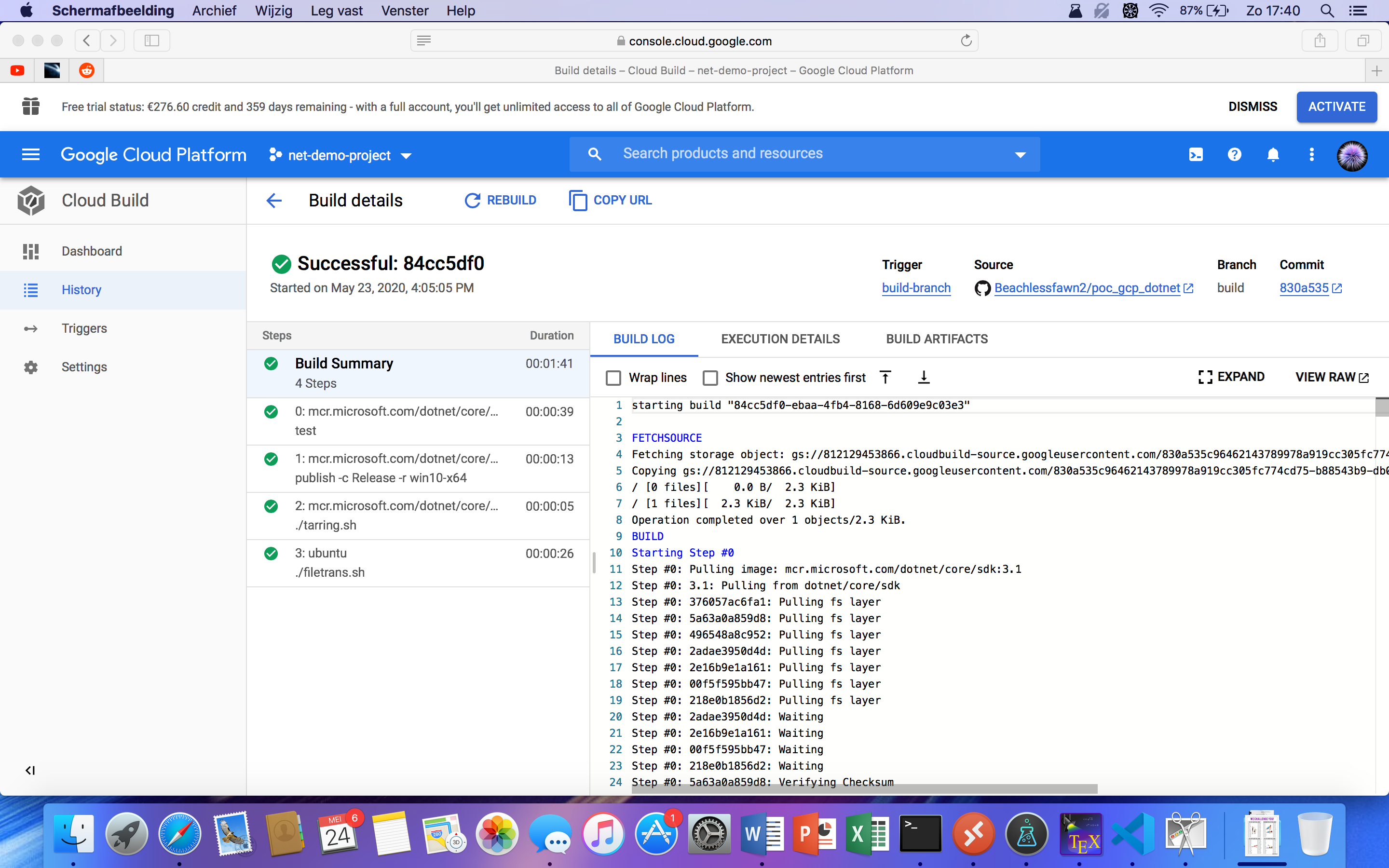
#! /bin/bash

cd /workspace/MessageUtil/bin/Release/netcoreapp3.1/win10-x64/ && tar -zcvf messageutil-win10-x64.tar.gz ./publish/

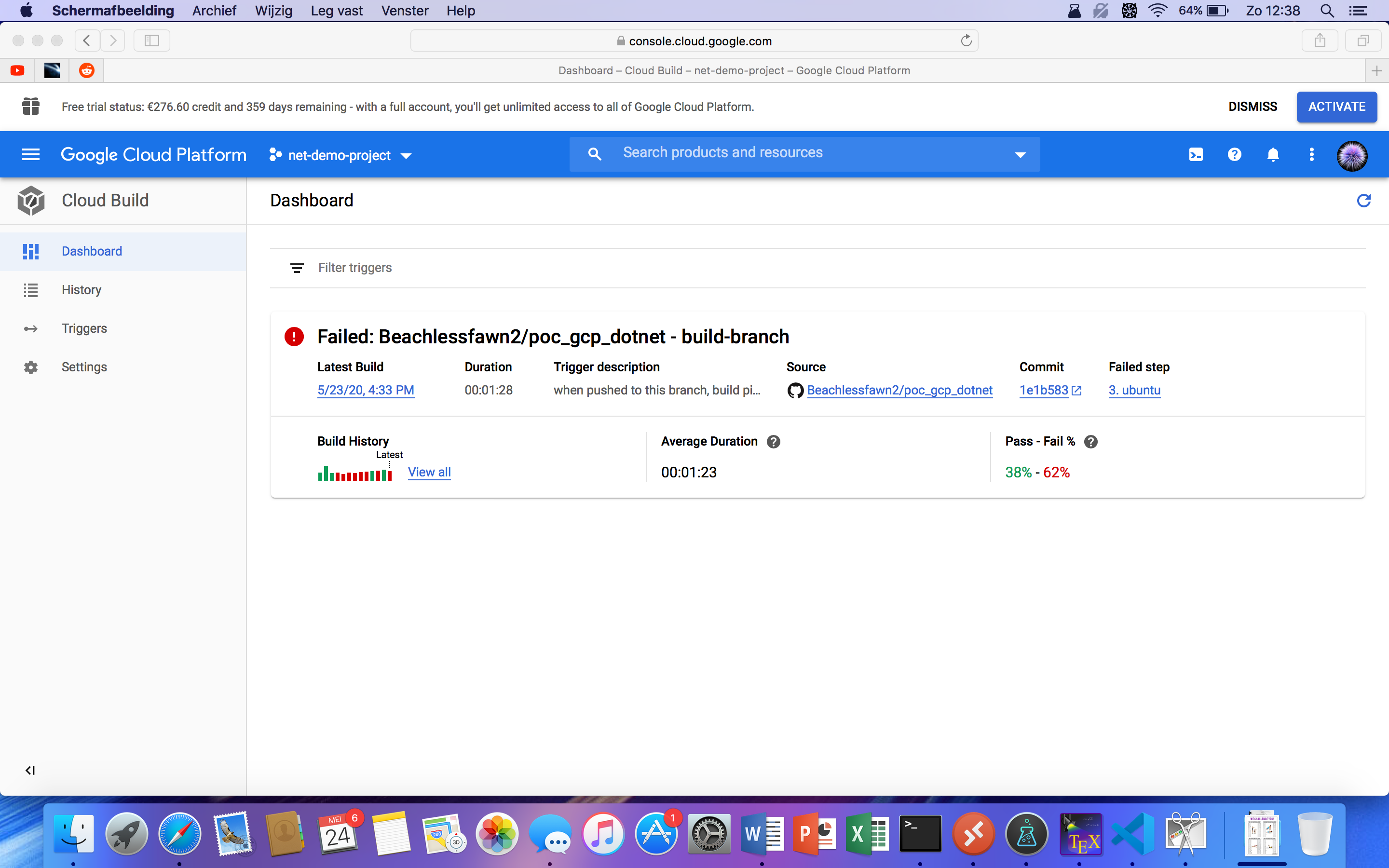
Nadat al deze bestanden aangemaakt geweest zijn, is deze pijpleiding voor het eerst uitgevoerd. Hiervoor is de Google Cloud CLI gebruikt op de gebruiker zijn computer. Het commando ‘gcloud builds submit’ gaat de volledige Git repositorie kopiëren naar een tijdelijke Storage Bucket en start vervolgens de pijpleiding op volgens de configuratie vanuit cloudbuild.yaml. Eenmaal dat deze pijpleiding volledig zonder fouten werkte is de Git repositorie op GitHib geplaatst. Vervolgens is er via de online console op Google Cloud een Trigger aangemaakt om deze pijpleiding automatisch te laten uitvoeren bij het updaten van een bepaalde tak op GitHub. Hiervoor is de GitHub gelinkt aan Google Cloud. Figuur (verwijzing) toont de gemaakte Trigger en ook de specifieke tag die gebruikt is voor de tak op GitHub. Er is gekozen om deze Trigger op een specifieke tak van de GitHub repositorie te configureren om het aantal keren dat deze pijpleiding uitgevoerd wordt te kunnen verminderen. Zo kunnen de ontwikkelaars zelf kiezen wanneer ze een build willen doen, door naar deze specifieke tak te committen. Dit vermijdt ook dat er in het wilde weg compilatie commando’s worden uitgevoerd door de ontwikkelaars zelf.



Eenmaal dat dit gebeurd is kon de trigger getest worden door een commit uit te voeren op GitHub. Vervolgens was het resultaat te testen op de lokale Windows Server 2019. Ook was er een gedetailleerd rapport te zien op de console van Google Cloud zelf. Om de applicatie uit te voeren moest simpel weg het gecomprimeerde bestand uitgepakt worden en uitgevoerd worden. Figuur (verwijzing) toont het resultaat van de uitgevoerde applicatie op de lokale machine. Figuur (verwijzing) toont de uitgevoerde compilatie op Google Cloud.



Tot slot is er op de online console van Google Cloud een simpel dashboard te zien. Dit dashboard toont de laatst uitgevoerde compilatie trigger van het project. Ook toont het eventuele fouten, gemiddelde tijden, wat de compilatie heeft doen uitvoeren, commit ID vanop GitHub, enz. Figuur (verwijzing) toont een voorbeeld van zo een dashboard.



Door deze POC op te stellen is er gebleken dat Google Cloud Build Code gemakkelijk te gebruiken is. Dat is, eenmaal de gebruiker er thuis in is. Er bestaat redelijk wat documentatie over wat allemaal mogelijk is in een ‘cloudbuild.yaml’. Helaas is voor een beginner het niet gemakkelijk om te verstaan hoe de verschillende containers met elkaar samenwerken en waar nu alle bestanden staan. Ook is een goede kennis van de containers nodig om te kunnen verstaan wat er allemaal mogelijk is. Dit zorgt ervoor dat het lang kan duren vooraleer er een pijpleiding operationeel is. Ook is het spijtig dat er niet meer analytics te verkrijgen zijn op Google Cloud Code Build. De gebruiker heeft enkel de analytics van wat er door Google zelf wordt aangeboden. Na het opstellen van deze POC is er ook gebleken dat Google Cloud niet zo duur is. Al het testen en uitproberen heeft slechts 0.40 euro gekost.

Het kan interessant zijn om in een later onderzoek te kijken of het niet mogelijk is om de gegenereerde log bestanden van Google Cloud te downloaden en te gebruiken voor eigen visualisatie systemen zoals: Grafana.