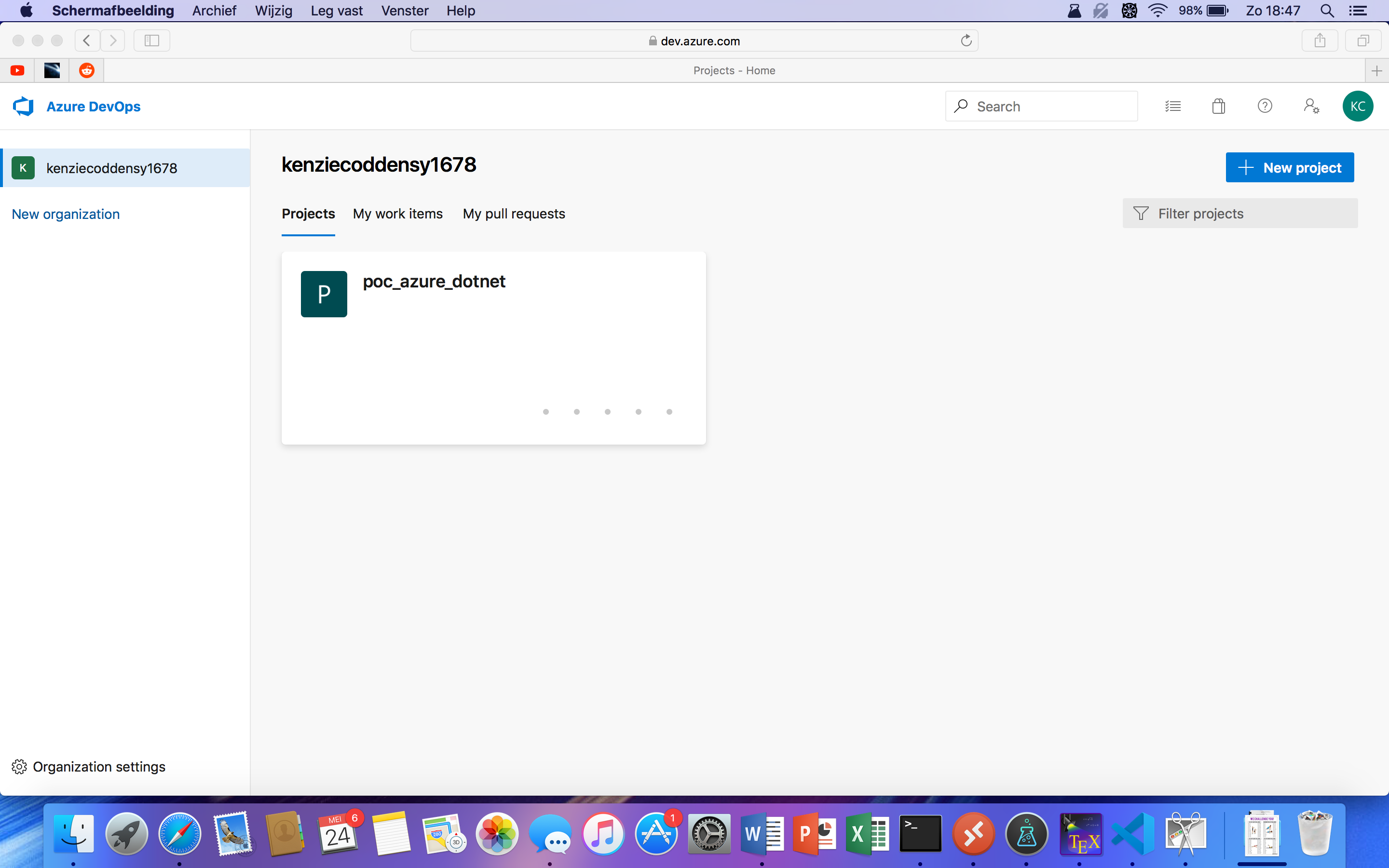
Naast de POC op Google Cloud is het ook nog interessant om dezelfde opstelling eens te maken in Azure DeVops. Dit om de ervaring en de functionaliteit naast elkaar te kunnen leggen. Voor de POC op Azure DeVops is er een nieuw DeVops project aangemaakt. Dit project heeft de naam ‘poc\_azure\_dotnet’ gekregen. Figuur (verwijzing) toont het aangemaakte project. Net zoals GCP bevatten projecten alle toegewezen producten en services die geconfigureerd geweest zijn. Ook is het gemakkelijk opnieuw te verwijderen en stopt dan ook de aanrekening.



Deze POC vertrekt vanuit dezelfde applicatie als bij GCP. Hiervoor 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 is er een gitignore aangemaakt. Figuur (verwijzing) toont een tree van de bestanden en mappen structuur. Deze repositorie is dan geüpload naar GitHub.

poc\_azure\_dotnet/

`--- .gitignore

`--- MessageUtil

   `--- MessageUtil.csproj

   `--- MessageUtilProgram.cs

`--- MessageUtil.sln

`--- MessageUtilTest

   `--- MessageUtilTest.csproj

   `--- MessageUtilTests.cs

Hierna is er op Azure DeVops een nieuwe pijpleiding aangemaakt. Dit was zeer gemakkelijk te volgen door middel van de online wizard. Eerst moest er gekozen worden wat versiebeheersysteem de gebruiker gebruikt. Hier is er gekozen voor GitHub. Vervolgens vraagt Azure DeVops de gebruiker toestemming voor het uitlezen van de repositories. Tegelijk wordt ook de Azure DeVops pijpleiding applicatie toegevoegd aan de gebruiker zijn GitHub. Na de initialisatie van de Azure DeVops applicatie voor GitHub moest er een reposirotrie geselecteerd worden. Hier is er gekozen voor de zojuist gemaakte GitHub repositorie. Azure DeVops leest deze repositorie uit en stelt dan op basis van de bestanden de juiste compilatie oplossing voor. Voor deze POC stelde Azure DeVops .Net Core voor als compilatie programma. Er is voor dezen oplossing gekozen. Azure DeVops toont hierna een Yaml-bestand, ‘azure-pipelines.yml’, met voor gemaakte stappen. Dit YAML-bestand bevat drie stappen. Een eerste stap die de juiste pakketten installeert op de virtuele machine. In een tweede stap wordt de applicatie gecompileerd. In de derde stap worden de meegeleverde testen uit figuur (verwijzing) uitgevoerd. Figuur (verwijzing) toont dit YAML-bestand.

Verwacht was dat dit een werkend vertrekpunt is voor de pijpleiding. Zeker omdat deze applicatie met een Microsoft specifieke taal gemaakt is, met behulp van het .Net framework dat ook Microsoft specifiek is. In eerste instantie leek deze pijpleiding te werken. Maar na het bekijken van de logbestanden is er vastgesteld dat de testen niet werden uitgevoerd. Er is gezocht naar een oplossing voor dit probleem. Er is geen duidelijke oplossing gevonden voor de huidige configuratie. De documentatie over deze modules was zeer onduidelijk en moeilijk te begrijpen. Zeker vanuit het standpunt van een beginner. In dit opzicht is er gekozen om opnieuw te beginnen aan de hand van een andere handleiding (verwijzing) op Azure. Deze gebruikt een voor gemaakte Windows virtuele machine met .Net core voor geïnstalleerd. Over deze module is uitgebreide documentatie te vinden. Daarnaast zijn er door Microsoft ook voorbeelden ter beschikking. Dit was dan ook het perfecte vertrekpunt.

Dit nieuw YAML-bestand bestaat uit zes stappen. In een eerste stap wordt de ‘MessageUtil’ applicatie volledige herbouwt. Deze staat dan klaar voor verdere stappen. In de tweede stap wordt de applicatie gecompileerd. De derde stap voert de meegeleverde testen uit. Ook publiceert het de resultaten op Azure DeVops. Figuur (verwijzing) toont een voorbeeld van uitgevoerde testen. De vierde stap compileert een uitrol versie voor het 64 bit Windows 10 platform in een map, publish. In de vijfde stap wordt er een PowerShell script uitgevoerd. Dit Script comprimeert de map publish. Figuur (verwijzing) toont het ‘zipping.ps1’ script. In een laatste stap wordt deze gecomprimeerde map gepubliceerd als een artifact zodanig dat de applicatie klaarstaat om in een latere stap te worden uitgerold. Figuur (verwijzing) toont het verbeterde ‘azure-pipelines.yml’. Vervolgens is deze code toegevoegd op GitHub.

Het CI-gedeelte van de pijpleiding werkt met deze code. Voor het CD-gedeelte moest er een nieuwe uitrol pijpleiding gemaakt worden. Op Azure DeVops is er een nieuwe release gemaakt bestaande uit een Linux Ubuntu machine. Deze virtuele machine download het gemaakte artifact en verzend dit bestand naar de lokale Windows Server. Deze virtuele machine voert het script uit figuur (verwijzing) uit. Het instellen van deze uitrol pijpleiding is zeer gemakkelijk door de online wizard. Figuur (verwijzing) toont het instellen van de virtuele machine. Ook zijn er talloze opties voor het instellen van toestemmingen, rechten, controles, enz. Dit kan interessant zijn in een productie omgeving. Deze POC laat dit buiten beschouwing. Figuur (verwijzing) toont de volledige pijpleiding.

Azure DeVops heeft ook zeer goede analytics. Voor deze POC zijn er een aantal voorbeelden gemaakt. Net zoal Google Cloud heeft Azure ook Dashboards. Figuur (verwijzing) toont een overzicht van het project.

Uit deze POC is gebleken dat Azure DeVops veel minder complex is om te configureren dan Google Cloud Code Build. Dit komt door de gemakkelijk te gebruiken wizards die de gebruiker door de configuratie heen gidsen. Zo voelt de complexiteit van het configureren van een pijpleiding minder zwaar aan. Uit deze POC is gebleken dat de standaardoplossing die Azure DeVops voorstelt niet altijd even gemakkelijk is om aan te passen naar de gebruiker zijn noden. Deze POC heeft ook aangetoond dat de documentatie van Azure DeVops redelijk complex kan zijn. Zelfs na meerdere projecten te hebben getest, zijn sommige zaken nog niet helemaal duidelijk. Azure DeVops biedt zeer uitgebreide hulpmiddelen aan om allerlei data te visualiseren. Deze dashboards zijn gemakkelijk te creëren. Ook zijn deze zeer duidelijk en overzichtelijk. Dit in tegenstelling tot Google Cloud. In vergelijking met Azure DeVops is Google Cloud gemakkelijker om aangepaste pijpleidingen te maken. Dit omdat het intuïtiever is om de configuratie YAML-bestanden te maken. Ook omdat Google Cloud Code Build met Docker containers werkt. Dit zorgt ervoor dat compileer machines veel uitgebreider kunnen aangepast worden naar de noden van de gebruiker.

Benaderd vanuit het Microsoft ecosysteem is Azure DeVops een zeer goede en logische keuze. Zeker omdat een hele hoop producten en services, die anders aanvullende kosten hebben, vrij te gebruiken zijn. Ook zijn alle hulpmiddelen om software te ontwikkelen volledig geïntegreerd in het platform. Gebruikers die buiten dit ecosysteem werken kijken beter naar Google Cloud in samenwerking met andere tools en oplossingen.