Voor deze opstelling is deze handleiding (verwijzing) gebruikt geweest. Vooraleer deze handleiding kon gevolgd worden, is er eerst een nieuwe organisatie aangemaakt geweest in Azure DeVops. Binnen deze organisatie is er dan een nieuw project aangemaakt geweest om de pijpleidingen in te configureren. Zo een Azure DeVops project bevat een hele serie producten en services om ontwikkelaars en organisaties te helpen in het softwareontwikkelingsproces. Ook is zo een project handig omdat het net zoals de andere Cloud platformen, alle aanrekeningen en producten stopzet bij het verwijderen.

Verder moest er ook een GitHub repositorie bestaan om deze handleiding te kunnen aanvatten. Deze repositorie bevat de Java applicatie met de hoofdklasse en de testklasse. Figuur (verwijzing) en figuur (verwijzing) toont de code van deze Java klassen. Ook moet er een pom.xml aanwezig zijn. Figuur (verwijzing) toont dit YAML-bestand. Deze bestanden zijn dan geüpload geweest naar GitHub.

Voor het aanmaken van een pijpleiding op Azure DeVops is de documentatie verder gevold. De combinatie van de handleiding en de online wizards maakten het configureren van een pijpleiding zeer gemakkelijk. In een eerste stap kiest de gebruiker de bron van de code. Hier is GitHub gekozen, zoals de handleiding aanbeval. Dan werd er automatisch omgeleid naar de GitHub login pagina. Hier kan de gebruiker zijn login gegevens ingeven. Bij het inloggen wordt de gebruiker gevraagd om de Azure DeVops applicatie toe te voegen aan GitHub. Deze applicatie staat Azure DeVops toe om rechtstreeks met de broncode te kunnen interacten. Ook worden er automatisch triggers ingesteld. Deze kunnen later worden aangepast naar de gebruiker zijn noden. Nadat de GitHub applicatie was toegevoegd, werd de broncode uitgelezen door Azure DeVops. Azure DeVops stelt op basis hiervan, de beste optie voor het compileren van de code voor. In dit geval was dat Maven.

Dan gaat Azure DeVops automatisch een YAML-bestand gaan toevoegen aan de gebruiker zijn broncode op GitHub. Dit YAML-Bestand heeft de naam ‘azure-pipelines.yml’ en bevat alle configuratie in verband met het CI gedeelte van de pijpleiding. Figuur (verwijzing) toont de code van het gegenereerde YAML-bestand in dit geval. In dit bestand staan er een aantal zaken gedefinieerd. Hier kan de gebruiker een trigger gaan instellen voor de pijpleiding. Hier is dit de master tak. Verder kan de gebruiker een virtuele machine definiëren. Tot slot worden de compileer stappen gedefinieerd. Hier zijn er twee stappen gedefinieerd geweest. In de eerste stap wordt de volledige applicatie gecompileerd. Ook worden de meegeleverde Junit testen uitgevoerd. En de applicatie wordt gecompileerd tot een Jar-bestand. In de tweede stap wordt het Jar-bestand geupload naar Azure DeVops. Deze laatste stap staat niet in de handleiding. Het opzoeken van de nodige informatie voor deze stap te definiëren was niet tijdrovend. De oplossing was snel gevonden in andere documentatie die meer uitleg geeft over Azure DeVops Artifacts. Ook moet voor deze stap niks extra toegevoegd worden aan het project. Azure DeVops creëert automatisch de gewenste bestanden structuur.

# Maven

# Build your Java project and run tests with Apache Maven.

# Add steps that analyze code, save build artifacts, deploy, and more:

# https://docs.microsoft.com/azure/devops/pipelines/languages/java

trigger:

- master

pool:

vmImage: 'ubuntu-latest'

steps:

- task: Maven@3

inputs:

mavenPomFile: 'pom.xml'

mavenOptions: '-Xmx3072m'

javaHomeOption: 'JDKVersion'

jdkVersionOption: '1.8'

jdkArchitectureOption: 'x64'

publishJUnitResults: true

testResultsFiles: '\*\*/surefire-reports/TEST-\*.xml'

goals: 'package'

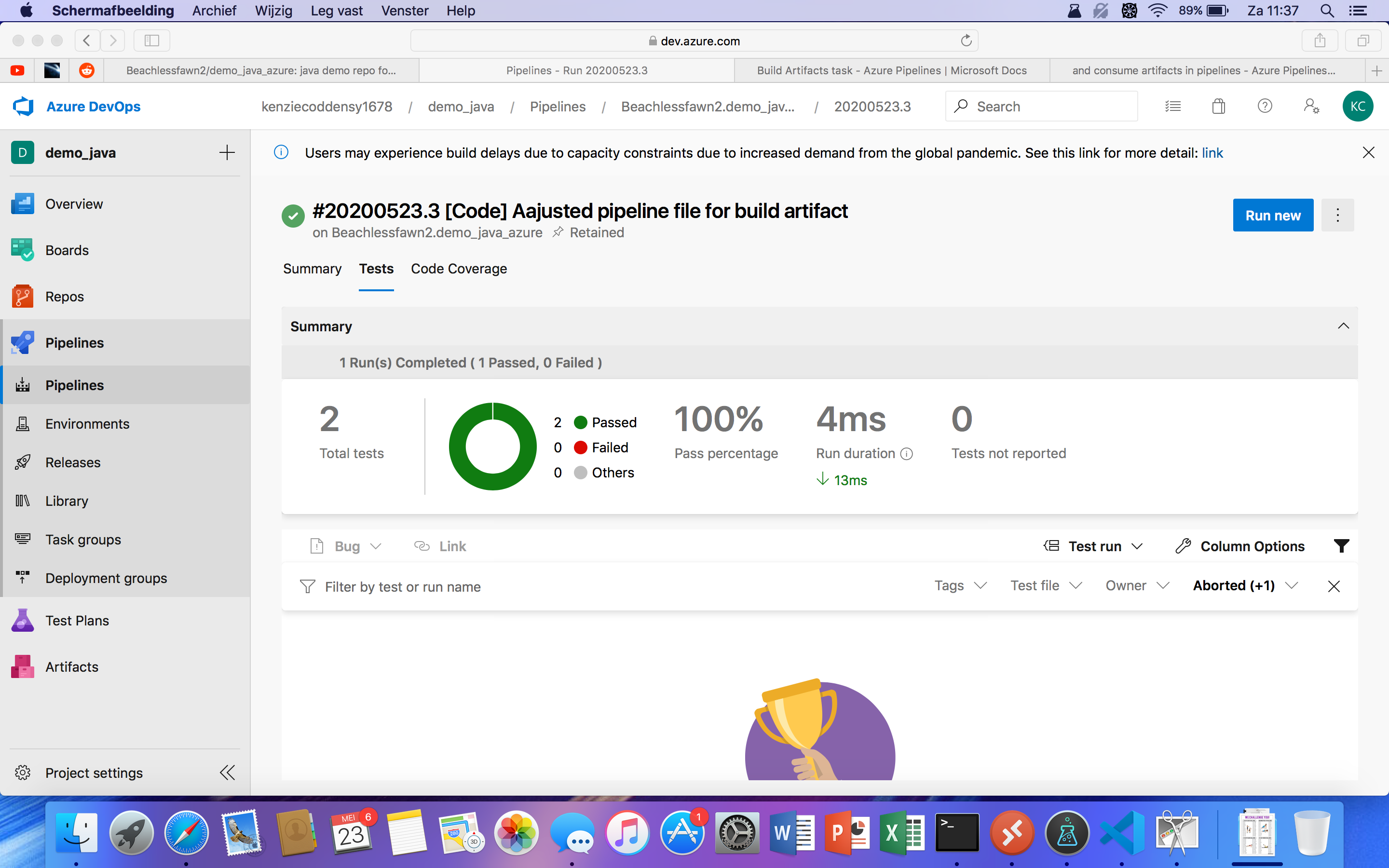
- task: PublishPipelineArtifact@1

inputs:

targetPath: $(System.DefaultWorkingDirectory)/target/messageUtil-1.0.jar

artifactName: messageUtil

Nadat dit YAML-bestand toegevoegd geweest is aan de broncode op GitHub, is de pijpleiding uitvoerbaar. Azure DeVops visualiseert de resultaten van het uitvoeren van deze pijpleiding zeer goed. Figuur (verwijzing) toont het dashboard van de testen. Hierop kan de gebruiker zijn slaag percentages aflezen. Ook kan de gebruiker direct zien welke testen er gefaald zijn.



Deze handleiding is duidelijk geschreven. Een gebruiker kan deze perfect volgen zelfs als erop voor hand geen kennis is over het platform. Ook het configureren van de pijpleiding zelf is veel minder complex dan de andere Cloud platformen. Extra informatie en documentatie is gemakkelijk te vinden en deze is gemakkelijk te begrijpen. Het hele proces voelde zeer intuïtief aan. Bovenal, heeft het weinig tijd gekost om deze pijpleiding te configureren. De bestanden structuur voor de applicatie heeft langer geduurd om te maken dan de pijpleiding zelf. Tabel (verwijzing) toont de gespendeerde tijd van begin tot einde voor alle drie de kandidaten. Dit product van Azure is zeer krachtig en integreert goed met andere tools en bestaande procedures. Het zal dan ook niet gemakkelijk zijn voor het andere Cloud platform om dit te evenaren.