trigger:

- develop

- master

variables:

  - group: Vanquis-Mobile-App-common

  - group: Vanquis-Mobile-App-dev1

  - group: Vanquis-Mobile-App-tst1

name: '$(Date:yyyyMMdd)$(Rev:r)'

pool:

  vmImage: 'macOS-10.15'

stages:

- stage: Mobile\_Dev\_Stage

  jobs:

  - job: Build\_Flutter\_Dev

    steps:

      - template: Deployment/AzurePipelines/build-flutter.yml

  - job: Build\_Android\_Dev

    dependsOn: Build\_Flutter\_Dev

    steps:

    - template: Deployment/AzurePipelines/build-android.yml

      parameters:

        buildFlavour: $(flutter\_flavor\_dev)

        entryPoint: $(flutter\_entrypoint\_dev)

        debugMode: $(flutter\_debug\_dev)

  - job: Build\_iOS\_Dev

    dependsOn: Build\_Flutter\_Dev

    steps:

    - template: Deployment/AzurePipelines/build-ios.yml

      parameters:

        buildFlavour: $(flutter\_flavor\_dev)

        entryPoint: $(flutter\_entrypoint\_dev)

        debugMode: $(flutter\_debug\_dev)

- stage: Mobile\_Test\_Stage

  dependsOn: Mobile\_Dev\_Stage

  condition: and(succeeded(), in(variables['Build.SourceBranch'], 'refs/heads/develop', 'refs/heads/master'))

  jobs:

  - job: Build\_Android\_Test

    steps:

    - template: Deployment/AzurePipelines/build-android.yml

      parameters:

        buildFlavour: $(flutter\_flavor\_test)

        entryPoint: $(flutter\_entrypoint\_test)

        debugMode: $(flutter\_debug\_test)

    - task: AndroidSigning@3

      displayName: 'Flutter: Android Signing'

      inputs:

        apkFiles: '\*\*/build/app/outputs/flutter-apk/\*-debug.apk'

        apksign: true

        apksignerKeystoreFile: $(android\_keystore\_filename\_test)

        apksignerKeystorePassword: $(android\_keystore\_password\_test)

        apksignerKeystoreAlias: $(android\_keystore\_alias\_test)

        apksignerKeyPassword: $(android\_key\_password\_test)

        apksignerArguments: --out $(Build.SourcesDirectory)/$(flutter\_project\_directory)/build/app/outputs/flutter-apk/$(mobile\_app\_name)-$(flutter\_flavor\_test)-debug-$(android\_version)-$(Build.Buildnumber).apk

        zipalign: false

      condition: succeeded()

    - template: Deployment/AzurePipelines/publish-android.yml

      parameters:

        targetFolder: 'android/test'

steps:

- task: Bash@3

  displayName: 'Install latest azure-cli'

  inputs:

    targetType: 'inline'

    script: |

      sudo apt-get update

      sudo apt remove azure-cli -y && sudo apt autoremove -y

      sudo apt-get update

      sudo apt-get install ca-certificates curl apt-transport-https lsb-release gnupg

      curl -sL https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/microsoft.gpg > /dev/null

      AZ\_REPO=$(lsb\_release -cs)

      echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ $AZ\_REPO main" | sudo tee /etc/apt/sources.list.d/azure-cli.list

      sudo apt-get update

      sudo apt-get install azure-cli

- task: Bash@3

  displayName: 'Install PowerShell'

  inputs:

    targetType: 'inline'

    script: |

      sudo apt-get update

      # Install pre-requisite packages.

      sudo apt-get install -y wget apt-transport-https

      # Download the Microsoft repository GPG keys

      wget -q https://packages.microsoft.com/config/ubuntu/16.04/packages-microsoft-prod.deb

      # Register the Microsoft repository GPG keys

      sudo dpkg -i packages-microsoft-prod.deb

      # Update the list of packages after we added packages.microsoft.com

      sudo apt-get update

      # Install PowerShell

      sudo apt-get install -y powershell

trigger:

- none

#- develop

#- master

variables:

  - name: build\_reference\_number

    value: $[format('{0:yyyyMMddHHmmss}', pipeline.startTime)]

  - group: mobile-flutter-common

  - group: mobile-build-flutterpoc-dev

  - group: mobile-build-flutterpoc-aut

  - group: mobile-build-flutterpoc-uat

  - group: mobile-build-flutterpoc-prod

  - group: mobile-build-flutter-devices

name: '$(Build.DefinitionName)\_$(SourceBranchName)\_$(build\_reference\_number)'

pool:

  vmImage: 'macOS-10.15'

stages:

- stage: Build\_Development

  jobs:

  - job: Build\_Flutter\_Development

#    condition: eq(1,2)

    steps:

      - template: Deployment/AzurePipelines/build-flutter.yml

  - job: Build\_Android\_Development

    dependsOn: Build\_Flutter\_Development

#    condition: eq(1,2)

    condition: succeeded()

    steps:

    - template: Deployment/AzurePipelines/build-android.yml

      parameters:

        flavor: $(flutter\_flavor\_dev)

        entryPoint: $(flutter\_entrypoint\_dev)

  - job: Build\_iOS\_Development

    dependsOn: Build\_Flutter\_Development

#    condition: eq(1,2)

    condition: succeeded()

    steps:

    - template: Deployment/AzurePipelines/build-ios.yml

      parameters:

        flavor: $(flutter\_flavor\_dev)

        entryPoint: $(flutter\_entrypoint\_dev)

- stage: Update\_Build\_Number

  dependsOn: Build\_Development

  jobs:

  - job: Update\_Pubspec

    condition: and ( succeeded(), or( eq(variables['Build.Reason'], 'IndividualCI'), eq(variables['Build.Reason'], 'BatchedCI') ) )

    steps:

      - checkout: self

#        persistCredentials: true

        clean: true

        fetchDepth: 1

      - script: |

          perl -i -pe 's/^(version:\s+\d+\.\d+\.\d+\+)(\d+)$/$1.($2+1)/e' $FLUTTER\_PROJECT\_DIRECTORY/pubspec.yaml

          version=`grep 'version: ' $FLUTTER\_PROJECT\_DIRECTORY/pubspec.yaml | sed 's/version: //'`

          git commit -m "Bump version to $version" $FLUTTER\_PROJECT\_DIRECTORY/pubspec.yaml

          git -c http.extraheader="AUTHORIZATION: bearer $SYSTEM\_ACCESSTOKEN" push origin HEAD:$BUILD\_SOURCEBRANCHNAME -q

        displayName: 'Flutter: Update Build Number'

        workingDirectory: $(Build.SourcesDirectory)

        condition: succeeded()

        failOnStderr: true

        env:

          SYSTEM\_ACCESSTOKEN: $(System.AccessToken)

- stage: Build\_Automation

  dependsOn: Update\_Build\_Number

#  condition: eq(1,2)

  condition: and ( succeeded(), ne(variables['Build.Reason'], 'PullRequest') )

  jobs:

  - job: Build\_Android\_Automation

    steps:

    - template: Deployment/AzurePipelines/install-android-virtual-device.yml

      parameters:

        android\_virtual\_device\_name: $(android\_virtual\_device\_name)

        android\_package\_name: $(android\_package\_name)

    - template: Deployment/AzurePipelines/build-android.yml

      parameters:

        flavor: $(flutter\_flavor\_aut)

        entryPoint: $(flutter\_entrypoint\_aut)

    - task: AndroidSigning@3

      displayName: 'Flutter: Android Signing'

      inputs:

        apkFiles: '\*\*/build/app/outputs/apk/\*\*/\*.apk'

        apksign: 'true'

        apksignerKeystoreFile: $(android\_keystore\_filename\_aut)

        apksignerKeystorePassword: $(android\_keystore\_password\_aut)

        apksignerKeystoreAlias: $(android\_keystore\_alias\_aut)

        apksignerKeyPassword: $(android\_key\_password\_aut)

      condition: succeeded()

    - template: Deployment/AzurePipelines/run-flutter-driver-integration.yml

      parameters:

        virtual\_device\_name: $(flutter\_android\_device\_identifier)

    - template: Deployment/AzurePipelines/publish-android.yml

      parameters:

        targetFolder: 'android/aut'

  - job: Build\_iOS\_Automation

    #condition: eq(1,2)

    steps:

    - task: InstallAppleProvisioningProfile@1

      displayName: 'Install Apple provisioning profile'

      inputs:

        provProfileSecureFile: $(ios\_provisioning\_profile\_filename\_aut)

        removeProfile: true

    - task: InstallAppleCertificate@2

      displayName: 'Install Apple certificate'

      inputs:

        certSecureFile: $(ios\_signing\_certificate\_filename\_aut)

        certPwd: $(ios\_signing\_certificate\_password\_aut)

        deleteCert: true

    - template: Deployment/AzurePipelines/install-ios-virtual-device.yml

      parameters:

        ios\_virtual\_device\_name: $(ios\_virtual\_device\_name)

    - template: Deployment/AzurePipelines/build-ios.yml

      parameters:

        flavor: $(flutter\_flavor\_aut)

        entryPoint: $(flutter\_entrypoint\_aut)

    - template: Deployment/AzurePipelines/run-flutter-driver-integration.yml

      parameters:

        virtual\_device\_name: $(ios\_virtual\_device\_name)

- stage: Build\_UAT

  dependsOn: Build\_Automation

  condition: eq(1,2)

#  condition: and ( succeeded(), ne(variables['Build.Reason'], 'PullRequest') )

  jobs:

  - job: Build\_Android\_UAT

    steps:

    - template: Deployment/AzurePipelines/build-android.yml

      parameters:

        flavor: $(flutter\_flavor\_uat)

        entryPoint: $(flutter\_entrypoint\_uat)

    - task: AndroidSigning@3

      displayName: 'Flutter: Android Signing'

      inputs:

        apkFiles: '\*\*/build/app/outputs/apk/\*\*/\*.apk'

        apksign: 'true'

        apksignerKeystoreFile: $(android\_keystore\_filename\_uat)

        apksignerKeystorePassword: $(android\_keystore\_password\_uat)

        apksignerKeystoreAlias: $(android\_keystore\_alias\_uat)

        apksignerKeyPassword: $(android\_key\_password\_uat)

      condition: succeeded()

    - template: Deployment/AzurePipelines/publish-android.yml

      parameters:

        targetFolder: 'android/uat'

  - job: Build\_iOS\_UAT

    condition: eq(1,2)

    steps:

    - task: InstallAppleProvisioningProfile@1

      displayName: 'Install Apple provisioning profile'

      inputs:

        provProfileSecureFile: $(ios\_provisioning\_profile\_filename\_uat)

        removeProfile: true

    - task: InstallAppleCertificate@2

      displayName: 'Install Apple certificate'

      inputs:

        certSecureFile: $(ios\_signing\_certificate\_filename\_uat)

        certPwd: $(ios\_signing\_certificate\_password\_uat)

        deleteCert: true

    - template: Deployment/AzurePipelines/build-ios.yml

      parameters:

        flavor: $(flutter\_flavor\_uat)

        entryPoint: $(flutter\_entrypoint\_uat)

- stage: Build\_Production

  dependsOn: Build\_UAT

  condition: eq(1,2)

#  condition: and ( succeeded(), ne(variables['Build.Reason'], 'PullRequest') )

  jobs:

  - job: Build\_Android\_Production

    steps:

    - template: Deployment/AzurePipelines/build-android.yml

      parameters:

        flavor: $(flutter\_flavor\_prod)

        entryPoint: $(flutter\_entrypoint\_prod)

    - task: AndroidSigning@3

      displayName: 'Flutter: Android Signing'

      inputs:

        apkFiles: '\*\*/build/app/outputs/apk/\*\*/\*.apk'

        apksign: 'true'

        apksignerKeystoreFile: $(android\_keystore\_filename\_prod)

        apksignerKeystorePassword: $(android\_keystore\_password\_prod)

        apksignerKeystoreAlias: $(android\_keystore\_alias\_prod)

        apksignerKeyPassword: $(android\_key\_password\_prod)

      condition: succeeded()

    - template: Deployment/AzurePipelines/publish-android.yml

  - job: Build\_iOS\_Production

    steps:

    - task: InstallAppleProvisioningProfile@1

      displayName: 'Install Apple provisioning profile'

      inputs:

        provProfileSecureFile: $(ios\_provisioning\_profile\_filename\_prod)

        removeProfile: true

    - task: InstallAppleCertificate@2

      displayName: 'Install Apple certificate'

      inputs:

        certSecureFile: $(ios\_signing\_certificate\_filename\_prod)

        certPwd: $(ios\_signing\_certificate\_password\_prod)

        deleteCert: true

    - template: Deployment/AzurePipelines/build-ios.yml

      parameters:

        flavor: $(flutter\_flavor\_prod)

        entryPoint: $(flutter\_entrypoint\_prod)

resources:

  repositories:

    - repository: Pipelines.Docker

      type: git

      name: Pipelines.Docker

      ref: refs/tags/v0.0.6

    - repository: Pipelines.Shared

      type: git

      name: Pipelines.Shared

      ref: refs/tags/v0.0.13

    - repository: Pipelines.Terraform

      type: git

      name: Pipelines.Terraform

      ref: refs/tags/v0.0.2

trigger:

  branches:

    include:

      - master

      - develop

variables:

  - group: "VBL API Nonprod"

  - group: "ssh"

  - name: service\_name

    value: "vanquismobilebff"

  - name: service\_connection

    value: "Non-prod Container Regsitry"

  - name: terraform\_version

    value: "0.13.5"

stages:

  - stage: Build

    jobs:

      - job: BuildAndRunUnitTests

        pool:

          vmImage: "ubuntu-latest"

        steps:

          - task: DotNetCoreCLI@2

            displayName: Install dependencies

            inputs:

              command: 'restore'

              projects: 'src/\*\*/\*.csproj'

              feedsToUse: 'select'

              vstsFeed: '5cf8bbfc-13f1-4704-9aed-e7dd72243f8e/96d98ab7-4269-474e-97d1-b33ab242dfec'

              workingDirectory: "./"

          - task: DotNetCoreCLI@2

            displayName: Build

            inputs:

              command: 'build'

              projects: 'src/\*\*/\*.csproj'

              arguments: '--configuration Release --no-restore'

              workingDirectory: "./"

          - task: DotNetCoreCLI@2

            displayName: Run Unit Tests

            inputs:

              command: 'test'

              projects: 'src/\*\*/\*.csproj'

              arguments: '--configuration Release --no-build --verbosity minimal'

              testRunTitle: 'Unit Tests'

              workingDirectory: "./"

      - job: BuildAndPush

        pool:

          vmImage: "ubuntu-latest"

        steps:

          - template: build-push-dockerimage.yml@Pipelines.Docker

            parameters:

              image\_repo\_name: $(service\_name)

              path\_to\_dockerfile: src/Vanquis.Mobile.BFF.WebApi/Dockerfile

              service\_connection: $(service\_connection)

              push\_to\_acr: true

              build\_context: "./src"

              tag\_step: set-version-variable-from-changelog.yml@Pipelines.Shared

  - stage: Dev1

    dependsOn: Build

    condition: and (succeeded(), in(variables['Build.SourceBranch'], 'refs/heads/develop', 'refs/heads/master'))

    jobs:

      - template: Deploy/deploy-pipelines.yml

        parameters:

            environment: dev1

  - stage: Tst1

    dependsOn: Dev1

    condition: and (succeeded(), in(variables['Build.SourceBranch'], 'refs/heads/develop', 'refs/heads/master'))

    jobs:

      - deployment: DeployToTest

        environment: vanquis-mobile-bff-tst

      - template: Deploy/deploy-pipelines.yml

        parameters:

            environment: tst1

trigger:

- master

- develop

pool:

  vmImage: 'ubuntu-latest'

stages:

- stage: Test

  variables:

    - group: 'API Platform - Test Environment'

  jobs:

  - job: TestTerraform

    steps:

    - task: qetza.replacetokens.replacetokens-task.replacetokens@3

      displayName: 'Replace tokens in \*\*/\*.tfvars'

      inputs:

        rootDirectory: ./example

        targetFiles: '\*\*/\*.tfvars'

    - script: |

        terraform init

      workingDirectory: './example'

      displayName: 'Terraform init'

    - script: |

        terraform validate

      workingDirectory: './example'

      displayName: 'Terraform validate'

    - script: |

        terraform apply -auto-approve

      workingDirectory: './example'

      displayName: 'Terraform apply '

    - script: |

        terraform destroy -auto-approve

      workingDirectory: './example'

      displayName: 'Terraform destroy'

resources:

  repositories:

    - repository: Pipelines.Docker

      type: git

      name: Pipelines.Docker

      ref: refs/tags/v0.0.1

    - repository: Pipelines.Shared

      type: git

      name: Pipelines.Shared

      ref: refs/tags/v0.0.3

trigger:

  branches:

    include:

    - master

    - feature/167248-configure-pod-access-to-servicebus

pool:

  vmImage: 'ubuntu-latest'

variables:

  - group: 'VBL API Nonprod'

  - group: 'ssh'

  - group: 'dev1-APIPlatform-Outputs'

  - name: imageRepoName

    value: 'testapp'

  - name: serviceConnection

    value: 'Non-prod Container Regsitry'

  - name: acrName

    value: 'acrcorenpweuapi'

#  - name: acrUrl

#    value: 'acrcorenpweuapi.azurecr.io'

stages:

- stage: Build

  jobs:

  - job: buildTestapp

    steps:

      - template: build-push-dockerimage.yml@Pipelines.Docker

        parameters:

          imageRepoName: $(imageRepoName)

          pathToDockerfile: src/TestApp/Dockerfile

          serviceConnection: $(serviceConnection)

          pushToACR: true

          tagStep: set-version-variable-from-changelog.yml@Pipelines.Shared

  - job: pushHelmCharts

    dependsOn: buildTestapp

    condition: succeeded()

    steps:

    - template: set-version-variable-from-changelog.yml@Pipelines.Shared

    - task: HelmInstaller@0

      displayName: Install Helm version 3.3.1

      inputs:

        helmVersion: '3.3.1'

        checkLatestHelmVersion: false

        installKubectl: true

        kubectlVersion: '1.19.0'

        checkLatestKubectl: false

    - script: |

        # Replace helm chart version

        # ${test%\*.\*}

        versionChart='$(Vanquis.Version)'

        sed -i "s/#version#/${versionChart%\*.\*}.$(Build.BuildId)/g" src/helm.charts/Chart.yaml

        cat src/helm.charts/Chart.yaml

        # Package the helm charts

        helm package src/helm.charts/ --app-version $(Vanquis.Version).$(Build.BuildId)

        # Login to Azure

        az login --service-principal -u $(client\_id) -p $(client\_secret) --tenant $(tenant\_id)

        az account set --subscription $(subscription\_name)

        # Authenticate from Helm to ACR

        #echo $(client\_secret) | helm registry login $(acrUrl) --username $(client\_id) --password-stdin

        # Push to ACR

        az acr helm push -n acrcorenpweuapi $(imageRepoName)-\*.tgz

      displayName: Publish helm chart to ACR

- stage: Dev1

  displayName: Deploy on Dev1

  jobs:

  - job: deploy

    steps:

    - template: set-version-variable-from-changelog.yml@Pipelines.Shared

    - task: InstallSSHKey@0

      inputs:

        knownHostsEntry: '$(known\_hosts)'

        sshPublicKey: '$(public\_key)'

        sshKeySecureFile: 'id\_rsa'

    - script: |

        # Clone the repository

        echo "----> Cloning repository"

        git clone git@ssh.dev.azure.com:v3/vbl-core/Vanquis%20Engineering/flux-dev1

        # Update helm release yaml file with the version we have built

        cp src/helm.release/helmrelease.yaml flux-dev1/$(imageRepoName).yaml

        echo "----> Replacing values on the yaml file"

        versionChart='$(Vanquis.Version)'

        serviceBus='$(service\_bus\_shared\_key)'

        echo "\n ${serviceBus}"

        sed -i "s/#chartVersion#/${versionChart%\*.\*}.$(Build.BuildId)/g" flux-dev1/$(imageRepoName).yaml

        sed -i "s/#appVersion#/${versionChart}.$(Build.BuildId)/g" flux-dev1/$(imageRepoName).yaml

        sed -i "s/#appName#/$(imageRepoName)/g" flux-dev1/$(imageRepoName).yaml

        sed -i "s/#acrName#/$(acrName)/g" flux-dev1/$(imageRepoName).yaml

        sed -i "s/#namespace#/default/g" flux-dev1/$(imageRepoName).yaml

        cat flux-dev1/$(imageRepoName).yaml

        # Push the changes to the repository

        echo "----> Pushing the changes"

        cd flux-dev1

        git config --global user.email "diego.manso@vanquisbank.com"

        git config --global user.name "Diego Manso"

        git add -A

        git commit -m "Add version $(Vanquis.Version).$(Build.BuildId)"

        git push

      displayName: Git push changes

trigger:

  - develop

  - master

pool:

  vmImage: 'ubuntu-20.04'

resources:

  repositories:

  - repository: Pipelines.Docker

    type: git

    name: Pipelines.Docker

    ref: refs/tags/v0.0.6

  - repository: Pipelines.Shared

    type: git

    name: Pipelines.Shared

    ref: refs/tags/v0.0.12

  - repository: Pipelines.Terraform

    type: git

    name: Pipelines.Terraform

    ref: refs/tags/v0.0.2

variables:

  - group: 'VBL API Nonprod'

  - group: 'ssh'

  - group: 'Git Access Token'

  - group: 'VBL Commercial Engineering DEV TEST Terraform Remote State'

  - group: 'Vanquis.Acquisitions.BFF'

  - name: service\_name

    value: 'acquisitionsbff'

  - name: service\_connection

    value: 'Non-prod Container Regsitry'

  - name: terraform\_version

    value: '0.13.5'

  - name: buildConfiguration

    value: 'Release'

  - name: tag\_version

    value: '$(Vanquis.Version).$(Build.BuildId)'

stages:

  - stage: UnitTests

    displayName: Unit Testing

    variables:

    - name: buildConfiguration

      value: 'Debug'

    jobs:

    - job: UnitTesting

      pool:

        name: Commercial engineering pool 2 - Windows

      workspace:

        clean: all

      displayName: 'Unit Testing'

      steps:

      - template: Deployment/AzurePipelines/unit-tests-pipeline.yml

  - stage: IntegrationTests

    displayName: Integration Tests

    condition: |

      and(

        ne(variables['Build.Reason'], 'PullRequest'),

        in(variables['Build.SourceBranch'], 'refs/heads/develop', 'refs/heads/master')

      )

    variables:

    - name: buildConfiguration

      value: 'Debug'

    jobs:

    - job: IntegrationTestsJob

      pool:

        name: Commercial engineering pool 2 - Windows

      workspace:

        clean: all

      displayName: 'Integration Testing'

      steps:

      - template: Deployment/AzurePipelines/integration-tests-pipeline.yml

  - stage: Build

    dependsOn: IntegrationTests

    condition: |

      and(

        ne(variables['Build.Reason'], 'PullRequest'),

        eq(variables['Build.SourceBranch'], 'refs/heads/develop')

      )

    displayName: Build Container and TF Artifacts

    jobs:

    - job: BuildBackend

      displayName: Build and Push Backend Container

      steps:

      - template: build-push-dockerimage.yml@Pipelines.Docker

        parameters:

          service\_name: $(service\_name)

          image\_repo\_name: $(service\_name)

          path\_to\_dockerfile: Dockerfile

          service\_connection: $(service\_connection)

          push\_to\_acr: true

          build\_context: '.'

          tag\_step: set-version-variable-from-changelog.yml@Pipelines.Shared

    - job: CopytoArtifactsAPIPlatformFiles

      displayName: Copy TF Artifacts

      steps:

      - task: CopyFiles@2

        displayName: "Copy helmrelease.yaml"

        inputs:

          Contents: 'Deployment/\*\*'

          TargetFolder: '$(Build.ArtifactStagingDirectory)'

      - task: CopyFiles@2

        displayName: "Copy helmrelease to /Deploy"

        inputs:

           SourceFolder: '$(Build.ArtifactStagingDirectory)/Deployment/APIPlatformFiles'

           Contents: '\*\*/\*'

           TargetFolder: '$(Build.ArtifactStagingDirectory)/Deploy'

      - task: CopyFiles@2

        displayName: "Copy ChangeLog.md"

        inputs:

          Contents: 'ChangeLog.md'

          TargetFolder: '$(Build.ArtifactStagingDirectory)'

      - task: PublishBuildArtifacts@1

        inputs:

          PathtoPublish: '$(Build.ArtifactStagingDirectory)'

          ArtifactName: 's'

          publishLocation: 'Container'

  - stage: Dev1

    dependsOn: Build

    condition: |

      and(

        ne(variables['Build.Reason'], 'PullRequest'),

        eq(variables['Build.SourceBranch'], 'refs/heads/develop')

      )

    displayName: Deploy on DEV API Platform

    variables:

    - group: 'dev1-APIPlatform-Variables'

    - group: 'dev1-APIPlatform-Outputs'

    - name: namespace

      value: 'default'

    - name: tfstate

      value: '$(service\_name).tfstate'

    - name: subnetname

      value: 'aks'

    - name: service\_type

      value: 'loans'

    - name: terraform\_version

      value: '0.13.5'

    - name: deploymentenvironment

      value: 'dev'

    jobs:

      - deployment: DeploytoAzure

        displayName: Deploy to Azure

        environment: Acquisitions BFF DEV

        strategy:

          runOnce:

            deploy:

              steps:

                - template: set-version-variable-from-changelog.yml@Pipelines.Shared

                - template: terraform-cd.yml@Pipelines.Terraform

                  parameters:

                    sshPublicKey: '$(public\_key)'

                    sshKnownHosts: '$(known\_hosts)'

                    sshKeySecureFile: 'id\_rsa'

                    subscriptionName: '$(subscription\_name)'

                    resourceGroupName: '$(resource\_group\_name)'

                    storageAccountName: '$(storage\_account\_name)'

                    containerName: '$(container\_name)'

                    workingDir: 'Deployment/Terraform'

                    tfstateKey: '$(tfstate)'

                    terraformVersion: '$(terraform\_version)'

                - template: Deployment/AzurePipelines/powershell-for-linux.yml

                - template: Deployment/AzurePipelines/deploy-to-aks.yml

                  parameters:

                    service\_name: $(service\_name)

                    known\_hosts: $(known\_hosts)

                    public\_key: $(public\_key)

                    subscription\_name: $(subscription\_name)

                    platform\_resource\_group: '$(platform\_resource\_group)'

                    aks\_cluster\_name: '$(aks\_cluster\_name)'

                    apim\_name: $(apim\_name)

  - stage: Tst1

    dependsOn: Dev1

    condition: |

      and(

        ne(variables['Build.Reason'], 'PullRequest'),

        eq(variables['Build.SourceBranch'], 'refs/heads/develop')

      )

    displayName: Deploy on TEST API Platform

    variables:

    - group: 'tst1-APIPlatform-Variables'

    - group: 'tst1-APIPlatform-Outputs'

    - name: namespace

      value: 'default'

    - name: tfstate

      value: '$(service\_name).tfstate'

    - name: service\_type

      value: 'loans'

    - name: subnetname

      value: 'aks'

    - name: terraform\_version

      value: '0.13.5'

    - name: deploymentenvironment

      value: 'tst'

    jobs:

      - deployment: DeploytoAzure

        displayName: Deploy to Azure

        environment: Acquisitions BFF TEST

        strategy:

          runOnce:

            deploy:

              steps:

                - template: set-version-variable-from-changelog.yml@Pipelines.Shared

                - template: terraform-cd.yml@Pipelines.Terraform

                  parameters:

                    sshPublicKey: '$(public\_key)'

                    sshKnownHosts: '$(known\_hosts)'

                    sshKeySecureFile: 'id\_rsa'

                    subscriptionName: '$(subscription\_name)'

                    resourceGroupName: '$(resource\_group\_name)'

                    storageAccountName: '$(storage\_account\_name)'

                    containerName: '$(container\_name)'

                    workingDir: 'Deployment/Terraform'

                    tfstateKey: '$(tfstate)'

                    terraformVersion: '$(terraform\_version)'

                - template: Deployment/AzurePipelines/powershell-for-linux.yml

                - template: Deployment/AzurePipelines/deploy-to-aks.yml

                  parameters:

                    service\_name: $(service\_name)

                    known\_hosts: $(known\_hosts)

                    public\_key: $(public\_key)

                    subscription\_name: $(subscription\_name)

                    platform\_resource\_group: '$(platform\_resource\_group)'

                    aks\_cluster\_name: '$(aks\_cluster\_name)'

                    apim\_name: $(apim\_name)

  - stage: Uat1

    dependsOn: Tst1

    condition: |

      and(

        ne(variables['Build.Reason'], 'PullRequest'),

        eq(variables['Build.SourceBranch'], 'refs/heads/develop')

      )

    displayName: Deploy on UAT API Platform

    variables:

    - group: 'uat1-APIPlatform-Variables'

    - group: 'uat1-APIPlatform-Outputs'

    - name: namespace

      value: 'default'

    - name: tfstate

      value: '$(service\_name).tfstate'

    - name: service\_type

      value: 'loans'

    - name: subnetname

      value: 'aks'

    - name: terraform\_version

      value: '0.13.5'

    - name: deploymentenvironment

      value: 'uat'

    jobs:

      - deployment: DeploytoAzure

        displayName: Deploy to Azure

        environment: Acquisitions BFF UAT

        strategy:

          runOnce:

            deploy:

              steps:

                - template: set-version-variable-from-changelog.yml@Pipelines.Shared

                - template: terraform-cd.yml@Pipelines.Terraform

                  parameters:

                    sshPublicKey: '$(public\_key)'

                    sshKnownHosts: '$(known\_hosts)'

                    sshKeySecureFile: 'id\_rsa'

                    subscriptionName: '$(subscription\_name)'

                    resourceGroupName: '$(resource\_group\_name)'

                    storageAccountName: '$(storage\_account\_name)'

                    containerName: '$(container\_name)'

                    workingDir: 'Deployment/Terraform'

                    tfstateKey: '$(tfstate)'

                    terraformVersion: '$(terraform\_version)'

                - template: Deployment/AzurePipelines/powershell-for-linux.yml

                - template: Deployment/AzurePipelines/deploy-to-aks.yml

                  parameters:

                    service\_name: $(service\_name)

                    known\_hosts: $(known\_hosts)

                    public\_key: $(public\_key)

                    subscription\_name: $(subscription\_name)

                    platform\_resource\_group: '$(platform\_resource\_group)'

                    aks\_cluster\_name: '$(aks\_cluster\_name)'

                    apim\_name: $(apim\_name)

resources:

  repositories:

    - repository: Pipelines.Docker

      type: git

      name: Pipelines.Docker

      ref: refs/tags/v0.0.6

    - repository: Pipelines.Shared

      type: git

      name: Pipelines.Shared

      ref: refs/tags/v0.0.15

    - repository: Pipelines.Terraform

      type: git

      name: Pipelines.Terraform

      ref: refs/tags/v0.0.2

trigger:

  branches:

    include:

      - master

      - develop

variables:

  - group: "VBL API Nonprod"

  - group: "ssh"

  - name: service\_name

    value: "vanquiscardsregisteredcard"

  - name: service\_connection

    value: "Non-prod Container Regsitry"

  - name: terraform\_version

    value: "0.13.5"

stages:

  - stage: Build

    jobs:

      - job: BuildAndRunUnitAndArchitectureTests

        pool:

          vmImage: "ubuntu-latest"

        steps:

        - task: DotNetCoreCLI@2

          displayName: Install dependencies

          inputs:

            command: 'restore'

            projects: '\*\*/\*.csproj'

            feedsToUse: 'select'

            vstsFeed: '5cf8bbfc-13f1-4704-9aed-e7dd72243f8e/96d98ab7-4269-474e-97d1-b33ab242dfec'

            workingDirectory: "./src"

        - task: DotNetCoreCLI@2

          displayName: Build

          inputs:

            command: 'build'

            projects: '\*\*/\*.csproj'

            arguments: '--configuration Release --no-restore'

            workingDirectory: "./src"

        - task: DotNetCoreCLI@2

          displayName: Run Unit Tests

          inputs:

            command: 'test'

            projects: '\*\*/\*.csproj'

            arguments: '--configuration Release --no-build --verbosity minimal --filter UnitTests'

            testRunTitle: 'Unit Tests'

            workingDirectory: "./src"

        - task: DotNetCoreCLI@2

          displayName: Run Architecture Tests

          inputs:

            command: 'test'

            projects: '\*\*/\*.csproj'

            arguments: '--configuration Release --no-build --verbosity minimal --filter ArchTests'

            testRunTitle: 'Arch Tests'

            workingDirectory: "./src"

      - job: BuildAndRunIntegrationTests

        pool:

          vmImage: "ubuntu-latest"

        steps:

        - task: Bash@3

          displayName: Start a container instance

          inputs:

            targetType: 'inline'

            script: |

              # Write your commands here

              docker-compose -f docker-compose-integration-tests.yml -f docker-compute-integration-tests.overrides.yml up -d

              sleep 30

        - task: DotNetCoreCLI@2

          displayName: Install dependencies

          inputs:

            command: 'restore'

            projects: '\*\*/\*.csproj'

            feedsToUse: 'select'

            vstsFeed: '5cf8bbfc-13f1-4704-9aed-e7dd72243f8e/96d98ab7-4269-474e-97d1-b33ab242dfec'

            workingDirectory: "./src"

        - task: DotNetCoreCLI@2

          displayName: Build

          inputs:

            command: 'build'

            projects: '\*\*/\*.csproj'

            arguments: '--configuration Release --no-restore'

            workingDirectory: "./src"

        - task: DotNetCoreCLI@2

          displayName: Run IntegrationTests Tests

          inputs:

            command: 'test'

            projects: '\*\*/Vanquis.Cards.RegisteredCard.WebApi.IntegrationTests.csproj'

            arguments: '--configuration Release --no-build --verbosity minimal'

            testRunTitle: 'Integration Tests'

            workingDirectory: "./src"

        - task: Bash@3

          displayName: Stop the container instance

          inputs:

            targetType: 'inline'

            script: |

              # Write your commands here

              docker-compose -f docker-compose-integration-tests.yml -f docker-compute-integration-tests.overrides.yml down

      - job: BuildAndPush

        pool:

          vmImage: "ubuntu-latest"

        steps:

          - template: build-push-dockerimage.yml@Pipelines.Docker

            parameters:

              image\_repo\_name: $(service\_name)

              path\_to\_dockerfile: src/Vanquis.Cards.RegisteredCard.WebApi/Dockerfile

              service\_connection: $(service\_connection)

              push\_to\_acr: true

              build\_context: "./src"

              tag\_step: set-version-variable-from-changelog.yml@Pipelines.Shared

  - stage: Dev1

    displayName: Deploy on Dev1

    condition: and(succeeded(), in(variables['Build.SourceBranch'], 'refs/heads/develop', 'refs/heads/master'))

    variables:

      - group: "dev1-APIPlatform-Outputs"

      - group: "vanquis-cards-registeredcard-dev1"

      - name: namespace

        value: "default"

      - name: tfstate

        value: "$(service\_name).tfstate"

      - name: version

        value: "$(Vanquis.Version).$(Build.BuildId)"

    jobs:

      - job: deploy

        pool: API-Platform-Nonprod

        steps:

          - template: set-version-variable-from-changelog.yml@Pipelines.Shared

          - template: terraform-cd.yml@Pipelines.Terraform

            parameters:

              sshPublicKey: '$(public\_key)'

              sshKnownHosts: '$(known\_hosts)'

              sshKeySecureFile: 'id\_rsa'

              subscriptionName: '$(subscription\_name)'

              resourceGroupName: '$(resource\_group\_name)'

              storageAccountName: '$(storage\_account\_name)'

              containerName: '$(container\_name)'

              workingDir: 'Deploy/terraform'

              tfstateKey: '$(tfstate)'

              terraformVersion: '$(terraform\_version)'

          - task: Bash@3

            inputs:

              targetType: 'inline'

              script: |

                sudo apt-get update

                sudo apt remove azure-cli -y && sudo apt autoremove -y

                sudo apt-get update

                sudo apt-get install ca-certificates curl apt-transport-https lsb-release gnupg

                curl -sL https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/microsoft.gpg > /dev/null

                AZ\_REPO=$(lsb\_release -cs)

                echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ $AZ\_REPO main" | sudo tee /etc/apt/sources.list.d/azure-cli.list

                sudo apt-get update

                sudo apt-get install azure-cli

                sudo apt-get update

                # Install pre-requisite packages.

                sudo apt-get install -y wget apt-transport-https

                # Download the Microsoft repository GPG keys

                wget -q https://packages.microsoft.com/config/ubuntu/16.04/packages-microsoft-prod.deb

                # Register the Microsoft repository GPG keys

                sudo dpkg -i packages-microsoft-prod.deb

                # Update the list of packages after we added packages.microsoft.com

                sudo apt-get update

                # Install PowerShell

                sudo apt-get install -y powershell

          - template: deploy-to-aks.yml@Pipelines.Shared

            parameters:

              service\_name: $(service\_name)

              known\_hosts: $(known\_hosts)

              public\_key: $(public\_key)

              subscription\_name: $(subscription\_name)

              platform\_resource\_group: '$(platform\_resource\_group)'

              aks\_cluster\_name: '$(aks\_cluster\_name)'

              apim\_name: $(apim\_name)

trigger:

- master

- develop

pool:

  vmImage: 'ubuntu-latest'

stages:

- stage: Test

  variables:

    - group: 'Test Variables'

  jobs:

  - job: TestTerraform

    steps:

    - script: |

        terraform version

        terraform init

      workingDirectory: './example'

      displayName: 'Terraform init'

    - script: |

        terraform validate

      workingDirectory: './example'

      displayName: 'Terraform validate'

    - script: |

        terraform apply -var client\_id=$(client\_id) -var client\_secret=$(client\_secret) -var tenant\_id=$(tenant\_id) -var subscription\_id=$(subscription\_id) -auto-approve

      workingDirectory: './example'

      displayName: 'Terraform apply'

    - script: |

        terraform destroy -var client\_id=$(client\_id) -var client\_secret=$(client\_secret) -var tenant\_id=$(tenant\_id) -var subscription\_id=$(subscription\_id) -auto-approve

      workingDirectory: './example'

      displayName: 'Terraform destroy'

trigger:

  - master

  - develop

  - feature/\*

pool:

  vmImage: ubuntu-latest

stages:

  - stage: Test

    variables:

      - group: API Platform - Test Environment

    jobs:

      - job: TestTerraform

        steps:

          - task: qetza.replacetokens.replacetokens-task.replacetokens@3

            displayName: Replace tokens in \*\*/\*.tfvars

            inputs:

              rootDirectory: ./example

              targetFiles: '\*\*/\*.tfvars'

          - script: |

              cat terraform.tfvars

              echo '$(pat\_token)' | az devops login --org https://dev.azure.com/vbl-core/

              terraform init

            workingDirectory: ./example

            displayName: Terraform init

          - script: |

              terraform validate

            workingDirectory: ./example

            displayName: Terraform validate

          - script: |

              terraform apply -auto-approve

            workingDirectory: ./example

            displayName: Terraform apply

          - script: |

              terraform destroy -auto-approve

            workingDirectory: ./example

            displayName: Terraform destroy

resources:

  repositories:

    - repository: Pipelines.Shared

      type: git

      name: Pipelines.Shared

      ref: refs/tags/v0.0.15

trigger:

  branches:

    include:

    - main

stages:

- stage: BuildAndTest

  jobs:

  - job: BuildAndTest

    steps:

    - task: UseDotNet@2

      displayName: UseDotNet5

      inputs:

        version: '5.0.x'

        includePreviewVersions: false

    - task: DotNetCoreCLI@2

      displayName: Install dependencies

      inputs:

        command: 'restore'

        projects: '\*\*/Vanquis.Validation.Account.csproj'

        workingDirectory: "./src"

    - task: DotNetCoreCLI@2

      displayName: Build

      inputs:

        command: 'build'

        projects: '\*\*/Vanquis.Validation.Account.csproj'

        arguments: '--configuration Release --no-restore'

        workingDirectory: "./src"

    - task: DotNetCoreCLI@2

      displayName: Test

      inputs:

        command: 'test'

        projects: '\*\*/Vanquis.Validation.Account.Test.csproj'

        arguments: '--configuration Release --no-build --verbosity minimal'

        testRunTitle: 'Unit Tests'

        workingDirectory: "./src"

- stage: PushPackage

  variables:

  - name: packageVersion

    value: '$(Vanquis.Version).$(Build.BuildId)'

  jobs:

  - job: BuildAndPush

    steps:

      - task: UseDotNet@2

        inputs:

          version: '5.0.x'

          includePreviewVersions: false

      - template: set-version-variable-from-changelog.yml@Pipelines.Shared

      - template: publish-nuget-packages-to-feed.yml@Pipelines.Shared

        parameters:

          project\_path: './src/Vanquis.Validation.Account/Vanquis.Validation.Account.csproj'

          package\_version: $(packageVersion)

          package\_pre\_release: ${{ notIn(variables['Build.SourceBranchName'], 'main') }}

          package\_publish: true

          package\_name: 'Vanquis.Validation.Account'

          execute\_tests: false

trigger:

- master

parameters:

- name: repository\_name

  displayName: Repository Name

  type: string

  default: 'Repository.Example.Name'

- name: service\_name

  displayName: Service Name

  type: string

  default: 'examplename'

- name: path\_to\_dockerfile

  displayName: Path to the Dockerfile

  type: string

  default: 'src/Example.Folder/Dockerfile'

- name: variable\_group\_name

  displayName: Variable group name on the Library (without environment)

  type: string

  default: 'Example Template'

- name: bdd\_test\_folder

  displayName: BDD tests folder

  type: string

  default: './src/Example.Bdd.AutomationTests'

- name: bdd\_test\_csproj\_file

  displayName: BDD tests csproj file name

  type: string

  default: 'Example.AutomationTests.csproj'

- name: add\_to\_api\_manager

  displayName: Do you need API Manager?

  type: boolean

  default: false

- name: keyvault

  displayName: Do you need a keyvault?

  type: boolean

  default: false

- name: servicebus\_topics\_access

  displayName: Do you need acess to Service Bus Topics?

  type: boolean

  default: false

#- name: stg1

#  type: boolean

#  default: false

#- name: prd1

#  type: boolean

#  default: false

variables:

  - group: "ssh"

  - name: pipeline\_service\_name

    value: '${{ parameters.service\_name }}'

  - name: pipeline\_path\_to\_dockerfile

    value: ${{ parameters.path\_to\_dockerfile }}

  - name: pipeline\_variable\_group\_name

    value: '${{ parameters.variable\_group\_name }}'

  - name: pipeline\_bdd\_test\_folder

    value: '${{ parameters.bdd\_test\_folder }}'

  - name: pipeline\_bdd\_test\_csproj\_file

    value: '${{ parameters.bdd\_test\_csproj\_file }}'

  - name: pipeline\_add\_to\_api\_manager

    value: ${{ parameters.add\_to\_api\_manager }}

stages:

  - stage: CreatePipeline

    jobs:

      - job: Replace\_variables

        steps:

          - task: qetza.replacetokens.replacetokens-task.replacetokens@3

            displayName: 'Replace values on pipeline files'

            inputs:

              rootDirectory: './Pipeline'

              targetFiles: '\*\*/\*.yml'

          - script: |

              cd ./Pipeline

              cat azure-pipelines.yml

            displayName: 'azure-pipelines'

          - script: |

              cd ./Pipeline

              cat deploy-pipelines.yml

            displayName: 'deploy-pipelines'

          - script: |

              cd ./Pipeline

              cat helmrelease.yaml

            displayName: 'helmrelease.yaml'

          - task: InstallSSHKey@0

            displayName: Install ssh key

            inputs:

              knownHostsEntry: '$(known\_hosts)'

              sshPublicKey: '$(public\_key)'

              sshKeySecureFile: 'id\_rsa'

          - script: |

              # Clone the repository

              echo "----> Cloning repository"

              git clone git@ssh.dev.azure.com:v3/vbl-core/Vanquis%20Engineering/${{ parameters.repository\_name }}

              # Copy the updated files

              cat ./Pipeline/deploy-pipelines.yml

              cp ./Pipeline/helmrelease.yaml ${{ parameters.repository\_name }}/Deploy/

              cp ./Pipeline/deploy-pipelines.yml ${{ parameters.repository\_name }}/Deploy/

              cp ./Pipeline/azure-pipelines.yml ${{ parameters.repository\_name }}/

              cp -R ./terraform ${{ parameters.repository\_name }}/Deploy/

              # Push the changes to the repository

              echo "----> Pushing the changes"

              cd ${{ parameters.repository\_name }}

              git checkout develop

              git pull

              #git push origin --delete feature/create-pipeline

              git checkout -b feature/create-pipeline

              git config user.email "diego.manso@vanquisbank.com"

              git config user.name "Diego Manso"

              git add -A

              git commit -m "Add pipelines files"

              git push --set-upstream origin feature/create-pipeline

            displayName: 'Add pipeline changes'

          - script: |

              cp ./terraform\_modules/keyvault.tf ${{ parameters.repository\_name }}/Deploy/terraform/

              cat ./terraform\_modules/keyvault\_outputs.tf >> ${{ parameters.repository\_name }}/Deploy/terraform/outputs.tf

              cd ${{ parameters.repository\_name }}

              git add -A

              git commit -m "Add keyvault to terraform"

              git push

            condition: and(succeeded(), eq('${{ parameters.keyvault }}', true))

            displayName: 'Add keyvault to terraform'

          - script: |

              cp ./terraform\_modules/servicebus\_topic\_access.tf ${{ parameters.repository\_name }}/Deploy/terraform/

              cat ./terraform\_modules/servicebus\_topic\_access\_variables.tf >> ${{ parameters.repository\_name }}/Deploy/terraform/variables.tf

              cat ./terraform\_modules/servicebus\_topic\_access\_variables\_tfvars.tf >> ${{ parameters.repository\_name }}/Deploy/terraform/terraform.tfvars

              cd ${{ parameters.repository\_name }}

              git add -A

              git commit -m "Add access to service bus topics"

              git push

            condition: and(succeeded(), eq('${{ parameters.servicebus\_topics\_access }}', true))

            displayName: 'Add access to service bus topics'

parameters:

  service\_connection: ''

  image\_repo\_name: ''

  path\_to\_dockerfile: 'Dockerfile'

  tag\_step: ''

  push\_to\_acr: false

  build\_context: '\*\*'

  build\_arguments: '--build-arg FEED\_ACCESSTOKEN="$(personal\_token)"'

steps:

- template: '${{ parameters.tag\_step }}'

- task: Docker@2

  displayName: Login to ACR

  inputs:

    command: login

    containerRegistry: ${{ parameters.service\_connection }}

- task: Docker@2

  displayName: Build image from Dockerfile

  inputs:

    command: build

    repository: ${{ parameters.image\_repo\_name }}

    dockerFile: ${{ parameters.path\_to\_dockerfile }}

    tags: $(Vanquis.Version).$(Build.BuildId)

    buildContext: ${{ parameters.build\_context }}

    arguments: ${{ parameters.build\_arguments }}

- task: Docker@2

  displayName: Push image to ACR

  inputs:

    command: push

    repository: ${{ parameters.image\_repo\_name }}

    tags: $(Vanquis.Version).$(Build.BuildId)

  condition: and(succeeded(), eq('${{ parameters.push\_to\_acr }}', true))

resources:

  repositories:

    - repository: Pipelines.Docker

      type: git

      name: Pipelines.Docker

      ref: refs/tags/v0.0.6

    - repository: Pipelines.Shared

      type: git

      name: Pipelines.Shared

      ref: refs/tags/v0.0.16

    - repository: Pipelines.Terraform

      type: git

      name: Pipelines.Terraform

      ref: refs/tags/v0.0.2

trigger:

  branches:

    include:

      - master

      - develop

variables:

  - group: "VBL API Nonprod"

  - group: "ssh"

  - name: service\_name

    value: "loanpaymentcalculationapi"

  - name: service\_connection

    value: "Non-prod Container Regsitry"

  - name: terraform\_version

    value: "0.13.5"

stages:

  - stage: Build

    jobs:

      - job: BuildAndRunUnitAndArchitectureTests

        pool:

          vmImage: "ubuntu-latest"

        steps:

        - task: UseDotNet@2

          inputs:

            packageType: sdk

            version: '5.0.x'

            includePreviewVersions: false # Required for preview versions

        - task: DotNetCoreCLI@2

          displayName: Install dependencies

          inputs:

            command: 'restore'

            projects: '\*\*/\*.csproj'

            feedsToUse: 'select'

            vstsFeed: '5cf8bbfc-13f1-4704-9aed-e7dd72243f8e/96d98ab7-4269-474e-97d1-b33ab242dfec'

            workingDirectory: "./src"

        - task: DotNetCoreCLI@2

          displayName: Build

          inputs:

            command: 'build'

            projects: '\*\*/\*.csproj'

            arguments: '--configuration Release --no-restore'

            workingDirectory: "./src"

        - task: DotNetCoreCLI@2

          displayName: Run Unit Tests

          inputs:

            command: 'test'

            projects: '\*\*/\*.csproj'

            arguments: '--configuration Release --no-build --verbosity minimal --filter UnitTests'

            testRunTitle: 'Unit Tests'

            workingDirectory: "./src"

        - task: DotNetCoreCLI@2

          displayName: Run Architecture Tests

          inputs:

            command: 'test'

            projects: '\*\*/\*.csproj'

            arguments: '--configuration Release --no-build --verbosity minimal --filter ArchTests'

            testRunTitle: 'Arch Tests'

            workingDirectory: "./src"

      - job: BuildAndPush

        dependsOn:

        - BuildAndRunUnitAndArchitectureTests

        condition: |

          and(

            succeeded(),

            ne(variables['Build.Reason'], 'PullRequest')

          )

        pool:

          vmImage: "ubuntu-latest"

        steps:

          - template: build-push-dockerimage.yml@Pipelines.Docker

            parameters:

              image\_repo\_name: $(service\_name)

              path\_to\_dockerfile: src/Loan.PaymentCalculation.WebApi/Dockerfile

              service\_connection: $(service\_connection)

              push\_to\_acr: true

              build\_context: "./src"

              tag\_step: set-version-variable-from-changelog.yml@Pipelines.Shared

  - stage: Dev1

    dependsOn: Build

    pool: 'API-Platform-Nonprod'

    condition: |

      and(

        succeeded(),

        ne(variables['Build.Reason'], 'PullRequest')

      )

    jobs:

        - template: Deploy/deploy-pipelines.yml

          parameters:

              environment: dev1

  - stage: Tst1

    dependsOn: Dev1

    pool: 'API-Platform-Nonprod'

    condition: and(succeeded(), in(variables['Build.SourceBranch'], 'refs/heads/develop'))

    jobs:

      - deployment: DeployToAzure

        environment: Loan Payment Calculation TST1

      - template: Deploy/deploy-pipelines.yml

        parameters:

            environment: tst1

      - job: IntegrationTest

        dependsOn: deploy

        variables:

        - group: "Loan Payment Calculation Api Variables - tst1"

        steps:

          - task: UseDotNet@2

            inputs:

              packageType: sdk

              version: '5.0.x'

              includePreviewVersions: false

          - task: DotNetCoreCLI@2

            displayName: Install dependencies

            inputs:

              command: 'restore'

              projects: '\*\*/\*.csproj'

              feedsToUse: 'select'

              vstsFeed: '5cf8bbfc-13f1-4704-9aed-e7dd72243f8e/96d98ab7-4269-474e-97d1-b33ab242dfec'

              workingDirectory: "./src"

          - task: qetza.replacetokens.replacetokens-task.replacetokens@3

            displayName: 'Replace values on appsettings.json'

            inputs:

              rootDirectory: './src/Loan.PaymentCalculation.WebApi.IntegrationTests'

              targetFiles: '\*\*/appsettings.json'

          - task: DotNetCoreCLI@2

            displayName: Run IntegrationTests Tests

            inputs:

              command: 'test'

              projects: '\*\*/Loan.PaymentCalculation.WebApi.IntegrationTests.csproj'

              arguments: '--configuration Release --verbosity minimal'

              testRunTitle: 'Integration Tests'

              workingDirectory: "./src"

  - stage: Uat1

    dependsOn: Build

    pool: 'API-Platform-Nonprod'

    condition: and(succeeded(), in(variables['Build.SourceBranch'], 'refs/heads/master'))

    jobs:

       - deployment: DeployToAzure

         environment: Loan Payment Calculation UAT1

       - template: Deploy/deploy-pipelines.yml

         parameters:

            environment: uat1

  - stage: Stg1

    dependsOn: Uat1

    condition: and(succeeded(), in(variables['Build.SourceBranch'], 'refs/heads/master'))

    variables:

    - group: "VBL API NEU Prod"

    jobs:

      - deployment: DeployToAzure

        environment: 'Loan Payment Calculation STG1'

      - template: Deploy/deploy-pipelines.yml

        parameters:

          environment: stg1

          pool: API-Platform-Prod

          subscription: 'vbl-api-prod'

  - stage: Prd1

    dependsOn: Stg1

    condition: and(succeeded(), in(variables['Build.SourceBranch'], 'refs/heads/master'))

    variables:

      - group: "VBL API NEU Prod"

    jobs:

        - deployment: DeployToAzure

          environment: 'Loan Payment Calculation PRD1'

        - template: Deploy/deploy-pipelines.yml

          parameters:

            environment: prd1

            pool: API-Platform-Prod

            subscription: 'vbl-api-prod'

trigger:

  - develop

  - master

pool:

  vmImage: 'ubuntu-20.04'

resources:

  repositories:

  - repository: Pipelines.Docker

    type: git

    name: Pipelines.Docker

    ref: refs/tags/v0.0.6

  - repository: Pipelines.Shared

    type: git

    name: Pipelines.Shared

    ref: refs/tags/v0.0.13

  - repository: Pipelines.Terraform

    type: git

    name: Pipelines.Terraform

    ref: refs/tags/v0.0.2

variables:

  - group: 'VBL API Nonprod'

  - group: 'ssh'

  - group: 'Git Access Token'

  - group: 'VBL Commercial Engineering DEV TEST Terraform Remote State'

  - name: service\_name

    value: 'disbursementui'

  - name: service\_connection

    value: 'Non-prod Container Regsitry'

  - name: service\_connection\_prod

    value: 'Prod Container Registry'

  - name: terraform\_version

    value: '0.13.5'

  - name: buildConfiguration

    value: 'Release'

  - name: tag\_version

    value: '$(Vanquis.Version).$(Build.BuildId)'

stages:

  - stage: UnitTesting

    displayName: Unit Testing

    jobs:

    - job: UnitTesting

      pool:

        vmImage: 'windows-latest'

      displayName: 'Unit Testing'

      steps:

      - template: Deployment/AzurePipelines/unit-tests-pipeline.yml

  - stage: Build

    dependsOn: UnitTesting

    condition: |

      and(

        ne(variables['Build.Reason'], 'PullRequest'),

        eq(variables['Build.SourceBranch'], 'refs/heads/develop')

      )

    displayName: develop-Build Container and TF Artifacts

    jobs:

    - job: BuildBackend

      displayName: Build and Push Backend Container

      steps:

      - template: build-push-dockerimage.yml@Pipelines.Docker

        parameters:

          service\_name: $(service\_name)

          image\_repo\_name: $(service\_name)

          path\_to\_dockerfile: Dockerfile

          service\_connection: $(service\_connection)

          push\_to\_acr: true

          build\_context: '.'

          tag\_step: set-version-variable-from-changelog.yml@Pipelines.Shared

    - job: CopytoArtifactsAPIPlatformFiles

      displayName: Copy TF Artifacts

      steps:

      - task: CopyFiles@2

        displayName: "Copy helmrelease.yaml"

        inputs:

          Contents: 'Deployment/\*\*'

          TargetFolder: '$(Build.ArtifactStagingDirectory)'

      - task: CopyFiles@2

        displayName: "Copy helmrelease to /Deploy"

        inputs:

           SourceFolder: '$(Build.ArtifactStagingDirectory)/Deployment/APIPlatformFiles'

           Contents: '\*\*/\*'

           TargetFolder: '$(Build.ArtifactStagingDirectory)/Deploy'

      - task: CopyFiles@2

        displayName: "Copy ChangeLog.md"

        inputs:

          Contents: 'ChangeLog.md'

          TargetFolder: '$(Build.ArtifactStagingDirectory)'

      - task: PublishBuildArtifacts@1

        inputs:

          PathtoPublish: '$(Build.ArtifactStagingDirectory)'

          ArtifactName: 's'

          publishLocation: 'Container'

  - stage: BuildProd

    dependsOn: UnitTesting

    condition: |

      and(

        ne(variables['Build.Reason'], 'PullRequest'),

        eq(variables['Build.SourceBranch'], 'refs/heads/master')

      )

    displayName: master-Build Container and TF Artifacts

    jobs:

    - job: BuildBackend

      displayName: Build and Push Backend Container

      steps:

      - template: build-push-dockerimage.yml@Pipelines.Docker

        parameters:

          service\_name: $(service\_name)

          image\_repo\_name: $(service\_name)

          path\_to\_dockerfile: Dockerfile

          service\_connection: $(service\_connection\_prod)

          push\_to\_acr: true

          build\_context: '.'

          tag\_step: set-version-variable-from-changelog.yml@Pipelines.Shared

    - job: CopytoArtifactsAPIPlatformFiles

      displayName: Copy TF Artifacts

      steps:

      - task: CopyFiles@2

        displayName: "Copy helmrelease.yaml"

        inputs:

          Contents: 'Deployment/\*\*'

          TargetFolder: '$(Build.ArtifactStagingDirectory)'

      - task: CopyFiles@2

        displayName: "Copy helmrelease to /Deploy"

        inputs:

           SourceFolder: '$(Build.ArtifactStagingDirectory)/Deployment/APIPlatformFiles'

           Contents: '\*\*/\*'

           TargetFolder: '$(Build.ArtifactStagingDirectory)/Deploy'

      - task: CopyFiles@2

        displayName: "Copy ChangeLog.md"

        inputs:

          Contents: 'ChangeLog.md'

          TargetFolder: '$(Build.ArtifactStagingDirectory)'

      - task: PublishBuildArtifacts@1

        inputs:

          PathtoPublish: '$(Build.ArtifactStagingDirectory)'

          ArtifactName: 's'

          publishLocation: 'Container'

  - stage: Dev1

    dependsOn: Build

    condition: |

      and(

        ne(variables['Build.Reason'], 'PullRequest'),

        eq(variables['Build.SourceBranch'], 'refs/heads/develop')

      )

    displayName: Deploy on DEV API Platform

    variables:

    - group: 'dev1-APIPlatform-Variables'

    - group: 'dev1-APIPlatform-Outputs'

    - name: namespace

      value: 'default'

    - name: tfstate

      value: '$(service\_name).tfstate'

    - name: subnetname

      value: 'aks'

    - name: service\_type

      value: 'loans'

    - name: terraform\_version

      value: '0.13.5'

    - name: sql\_connection\_string

      value: 'Server=sqlloandevweuloans.database.windows.net,1433;Database=db-loan-weu-dev-loans;Persist Security Info=False;Connect Timeout=30;Encrypt=True;TrustServerCertificate=False'

    - name: deploymentenvironment

      value: 'dev'

    - name: emailsettings\_to

      value: 'zeeshan.tariq@vanquisbank.co.uk'

    jobs:

      - deployment: DeploytoAzure

        displayName: Deploy to Azure

        environment: Loans Disbursement UI Backend DEV

        strategy:

          runOnce:

            deploy:

              steps:

                - template: set-version-variable-from-changelog.yml@Pipelines.Shared

                - template: terraform-cd.yml@Pipelines.Terraform

                  parameters:

                    sshPublicKey: '$(public\_key)'

                    sshKnownHosts: '$(known\_hosts)'

                    sshKeySecureFile: 'id\_rsa'

                    subscriptionName: '$(subscription\_name)'

                    resourceGroupName: '$(resource\_group\_name)'

                    storageAccountName: '$(storage\_account\_name)'

                    containerName: '$(container\_name)'

                    workingDir: 'Deployment/Terraform'

                    tfstateKey: '$(tfstate)'

                    terraformVersion: '$(terraform\_version)'

                - template: Deployment/AzurePipelines/powershell-for-linux.yml

                - template: Deployment/AzurePipelines/deploy-to-aks.yml

                  parameters:

                    service\_name: $(service\_name)

                    known\_hosts: $(known\_hosts)

                    public\_key: $(public\_key)

                    subscription\_name: $(subscription\_name)

                    platform\_resource\_group: '$(platform\_resource\_group)'

                    aks\_cluster\_name: '$(aks\_cluster\_name)'

                    apim\_name: $(apim\_name)

                - template: Deployment/AzurePipelines/cors-apim-fix.yml

                  parameters:

                    subscriptionName: '$(subscription\_name)'

                    tfstateKey: 'apimpolicy-$(tfstate)'

                    resourceGroupName: '$(resource\_group\_name)'

                    storageAccountName: '$(storage\_account\_name)'

                    containerName: '$(container\_name)'

  - stage: Tst1

    dependsOn: Dev1

    condition: |

      and(

        ne(variables['Build.Reason'], 'PullRequest'),

        eq(variables['Build.SourceBranch'], 'refs/heads/develop')

      )

    displayName: Deploy on TEST API Platform

    variables:

    - group: 'tst1-APIPlatform-Variables'

    - group: 'tst1-APIPlatform-Outputs'

    - name: namespace

      value: 'default'

    - name: tfstate

      value: '$(service\_name).tfstate'

    - name: service\_type

      value: 'loans'

    - name: subnetname

      value: 'aks'

    - name: terraform\_version

      value: '0.13.5'

    - name: sql\_connection\_string

      value: 'Server=sqlloandevweuloans.database.windows.net,1433;Database=db-loan-weu-dev-loans;Persist Security Info=False;Connect Timeout=30;Encrypt=True;TrustServerCertificate=False'

    - name: deploymentenvironment

      value: 'tst'

    - name: emailsettings\_to

      value: 'Benjamin.Cockle@vanquisbank.co.uk'

    jobs:

      - deployment: DeploytoAzure

        displayName: Deploy to Azure

        environment: Loans Disbursement UI Backend TEST

        strategy:

          runOnce:

            deploy:

              steps:

                - template: set-version-variable-from-changelog.yml@Pipelines.Shared

                - template: terraform-cd.yml@Pipelines.Terraform

                  parameters:

                    sshPublicKey: '$(public\_key)'

                    sshKnownHosts: '$(known\_hosts)'

                    sshKeySecureFile: 'id\_rsa'

                    subscriptionName: '$(subscription\_name)'

                    resourceGroupName: '$(resource\_group\_name)'

                    storageAccountName: '$(storage\_account\_name)'

                    containerName: '$(container\_name)'

                    workingDir: 'Deployment/Terraform'

                    tfstateKey: '$(tfstate)'

                    terraformVersion: '$(terraform\_version)'

                - template: Deployment/AzurePipelines/powershell-for-linux.yml

                - template: Deployment/AzurePipelines/deploy-to-aks.yml

                  parameters:

                    service\_name: $(service\_name)

                    known\_hosts: $(known\_hosts)

                    public\_key: $(public\_key)

                    subscription\_name: $(subscription\_name)

                    platform\_resource\_group: '$(platform\_resource\_group)'

                    aks\_cluster\_name: '$(aks\_cluster\_name)'

                    apim\_name: $(apim\_name)

                - template: Deployment/AzurePipelines/cors-apim-fix.yml

                  parameters:

                    subscriptionName: '$(subscription\_name)'

                    tfstateKey: 'apimpolicy-$(tfstate)'

                    resourceGroupName: '$(resource\_group\_name)'

                    storageAccountName: '$(storage\_account\_name)'

                    containerName: '$(container\_name)'

  - stage: Uat1

    dependsOn: Tst1

    condition: |

      and(

        ne(variables['Build.Reason'], 'PullRequest'),

        eq(variables['Build.SourceBranch'], 'refs/heads/develop')

      )

    displayName: Deploy on UAT API Platform

    variables:

    - group: 'uat1-APIPlatform-Variables'

    - group: 'uat1-APIPlatform-Outputs'

    - name: namespace

      value: 'default'

    - name: tfstate

      value: '$(service\_name).tfstate'

    - name: service\_type

      value: 'loans'

    - name: subnetname

      value: 'aks'

    - name: terraform\_version

      value: '0.13.5'

    - name: sql\_connection\_string

      value: 'Server=sqlloanuatweuloandisb.database.windows.net,1433;Database=db-loan-weu-uat-loandisb;Persist Security Info=False;Connect Timeout=30;Encrypt=True;TrustServerCertificate=False'

    - name: deploymentenvironment

      value: 'uat'

    - name: emailsettings\_to

      value: 'VanquisBankLoans@vanquisbank.co.uk'

    jobs:

      - deployment: DeploytoAzure

        displayName: Deploy to Azure

        environment: Loans Disbursement UI Backend UAT

        strategy:

          runOnce:

            deploy:

              steps:

                - template: set-version-variable-from-changelog.yml@Pipelines.Shared

                - template: terraform-cd.yml@Pipelines.Terraform

                  parameters:

                    sshPublicKey: '$(public\_key)'

                    sshKnownHosts: '$(known\_hosts)'

                    sshKeySecureFile: 'id\_rsa'

                    subscriptionName: '$(subscription\_name)'

                    resourceGroupName: '$(resource\_group\_name)'

                    storageAccountName: '$(storage\_account\_name)'

                    containerName: '$(container\_name)'

                    workingDir: 'Deployment/Terraform'

                    tfstateKey: '$(tfstate)'

                    terraformVersion: '$(terraform\_version)'

                - template: Deployment/AzurePipelines/powershell-for-linux.yml

                - template: Deployment/AzurePipelines/deploy-to-aks.yml

                  parameters:

                    service\_name: $(service\_name)

                    known\_hosts: $(known\_hosts)

                    public\_key: $(public\_key)

                    subscription\_name: $(subscription\_name)

                    platform\_resource\_group: '$(platform\_resource\_group)'

                    aks\_cluster\_name: '$(aks\_cluster\_name)'

                    apim\_name: $(apim\_name)

                - template: Deployment/AzurePipelines/cors-apim-fix.yml

                  parameters:

                    subscriptionName: '$(subscription\_name)'

                    tfstateKey: 'apimpolicy-$(tfstate)'

                    resourceGroupName: '$(resource\_group\_name)'

                    storageAccountName: '$(storage\_account\_name)'

                    containerName: '$(container\_name)'

  # - stage: stg1

  #   # dependsOn: BuildProd

  #   # condition: |

  #   #   and(

  #   #     ne(variables['Build.Reason'], 'PullRequest'),

  #   #     eq(variables['Build.SourceBranch'], 'refs/heads/master')

  #   #   )

  #   displayName: Deploy on Staging API Platform

  #   variables:

  #   - group: 'neu-stg1-APIPlatform-Variables'

  #   - group: 'VBL API NEU Prod'

  #   - group: 'prd-Core-Outputs'

  #   - group: 'stg1-APIPlatform-Outputs'

  #   - name: namespace

  #     value: 'default'

  #   - name: tfstate

  #     value: '$(service\_name).tfstate'

  #   - name: service\_type

  #     value: 'loans'

  #   - name: subnetname

  #     value: 'aks'

  #   - name: terraform\_version

  #     value: '0.13.5'

  #   - name: sql\_connection\_string

  #     value: 'Server=blankserver,1433;Database=blankdatabase;Persist Security Info=False;Connect Timeout=30;Encrypt=True;TrustServerCertificate=False'

  #   - name: deploymentenvironment

  #     value: 'stg'

  #   jobs:

  #     - deployment: DeploytoAzure

  #       displayName: Deploy to Azure

  #       environment: Loans Disbursement UI Backend Staging

  #       strategy:

  #         runOnce:

  #           deploy:

  #             steps:

  #               - template: set-version-variable-from-changelog.yml@Pipelines.Shared

  #               - template: terraform-cd.yml@Pipelines.Terraform

  #                 parameters:

  #                   sshPublicKey: '$(public\_key)'

  #                   sshKnownHosts: '$(known\_hosts)'

  #                   sshKeySecureFile: 'id\_rsa'

  #                   subscriptionName: '$(subscription\_name)'

  #                   resourceGroupName: '$(resource\_group\_name)'

  #                   storageAccountName: '$(storage\_account\_name)'

  #                   containerName: '$(container\_name)'

  #                   workingDir: 'Deployment/Terraform'

  #                   tfstateKey: '$(tfstate)'

  #                   terraformVersion: '$(terraform\_version)'

  #               - template: Deployment/AzurePipelines/powershell-for-linux.yml

  #               - template: Deployment/AzurePipelines/deploy-to-aks.yml

  #                 parameters:

  #                   service\_name: $(service\_name)

  #                   known\_hosts: $(known\_hosts)

  #                   public\_key: $(public\_key)

  #                   subscription\_name: $(subscription\_name)

  #                   platform\_resource\_group: '$(platform\_resource\_group)'

  #                   aks\_cluster\_name: '$(aks\_cluster\_name)'

  #                   apim\_name: $(apim\_name)

  #               - template: Deployment/AzurePipelines/cors-apim-fix.yml

  #                 parameters:

  #                   subscriptionName: '$(subscription\_name)'

  #                   tfstateKey: 'apimpolicy-$(tfstate)'

  #                   resourceGroupName: '$(resource\_group\_name)'

  #                   storageAccountName: '$(storage\_account\_name)'

  #                   containerName: '$(container\_name)'

  - stage: Production

    dependsOn: BuildProd

    condition: |

      and(

        ne(variables['Build.Reason'], 'PullRequest'),

        eq(variables['Build.SourceBranch'], 'refs/heads/master')

      )

    displayName: Deploy on Production API Platform

    variables:

    - group: 'neu-prd1-APIPlatform-Variables'

    - group: 'VBL API NEU Prod'

    - group: 'prd-Core-Outputs'

    - group: 'prd1-APIPlatform-Outputs'

    - name: namespace

      value: 'default'

    - name: tfstate

      value: '$(service\_name).tfstate'

    - name: service\_type

      value: 'loans'

    - name: subnetname

      value: 'aks'

    - name: terraform\_version

      value: '0.13.5'

    - name: sql\_connection\_string

      value: 'Server=sqlloanprdweuloandisb.database.windows.net,1433;Database=db-loan-weu-prd-loandisb;Persist Security Info=False;Connect Timeout=30;Encrypt=True;TrustServerCertificate=False'

    - name: deploymentenvironment

      value: 'prd'

    jobs:

      - deployment: DeploytoAzure

        displayName: Deploy to Azure

        environment: Loans Disbursement UI Backend Production

        strategy:

          runOnce:

            deploy:

              steps:

                - template: set-version-variable-from-changelog.yml@Pipelines.Shared

                - template: terraform-cd.yml@Pipelines.Terraform

                  parameters:

                    sshPublicKey: '$(public\_key)'

                    sshKnownHosts: '$(known\_hosts)'

                    sshKeySecureFile: 'id\_rsa'

                    subscriptionName: '$(subscription\_name)'

                    resourceGroupName: '$(resource\_group\_name)'

                    storageAccountName: '$(storage\_account\_name)'

                    containerName: '$(container\_name)'

                    workingDir: 'Deployment/Terraform'

                    tfstateKey: '$(tfstate)'

                    terraformVersion: '$(terraform\_version)'

                - template: Deployment/AzurePipelines/powershell-for-linux.yml

                - template: Deployment/AzurePipelines/deploy-to-aks.yml

                  parameters:

                    service\_name: $(service\_name)

                    known\_hosts: $(known\_hosts)

                    public\_key: $(public\_key)

                    subscription\_name: $(subscription\_name)

                    platform\_resource\_group: '$(platform\_resource\_group)'

                    aks\_cluster\_name: '$(aks\_cluster\_name)'

                    apim\_name: $(apim\_name)

                - template: Deployment/AzurePipelines/cors-apim-fix.yml

                  parameters:

                    subscriptionName: '$(subscription\_name)'

                    tfstateKey: 'apimpolicy-$(tfstate)'

                    resourceGroupName: '$(resource\_group\_name)'

                    storageAccountName: '$(storage\_account\_name)'

                    containerName: '$(container\_name)'

-------------------------------------------------------------------------------------------------------------------------