# Pipeline YAML

## 1. set up triggering

Trigger:

Pr by branch and path

pr:

  branches:

    include:

    - main

    - feature/\*

  paths:

    include:

    - ./Code/ingestors

    - ./BuildPipelines/\*.Ingestors.yml

    exclude:

    - ./Code/backend

    - ./Code/frontend

## 2. set up parameters

These are the parameters needed to run the pipeline. If not supplied, the default value is used.

parameters:

- name: env

  displayName: Select Environment

  type: string

  default: Dev

  values:

  - Dev

  - Test

  - UAT

  - Prod

This env parameter is then used to determine which variable group to use

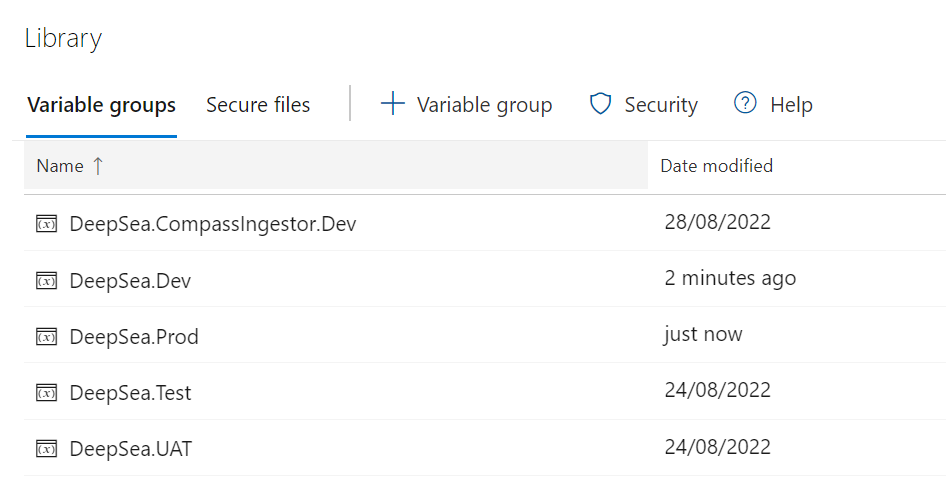
## 3. set up variables

This can be done at the pipeline level or at a job level.

variables:

  - group: DeepSea.DEV

Variable groups. Each group represents an environment.



In combination with environment parameter:

variables:

  - group: DeepSea.${{ parameters.env }}

This way, we select the variable group of the chosen environment

## 4. Job 1

Each job has properties and multiple tasks. Jobs are independently executable units.

* dependsOn – sets the dependency/order
* pool – selects build agent to run the job
* steps – contains all tasks in order

- job: BUILD\_AND\_TEST\_INGESTOR

    dependsOn: SONARCUBE

    displayName: Ingestor test and publish

    pool:

      name: QuantitativeTrading

      demands: ["docker"]

    steps:

      - script: echo '$(Build.BuildNumber)'

      - task: Bash@3

        displayName: Run docker unit tests

        inputs:

          filePath: "./Code/Ingestors/build\_and\_test.sh"

          arguments: "$(Common.TestResultsDirectory) unit"

          workingDirectory: "./Code/Ingestors/"

          failOnStderr: false

      - task: PublishTestResults@2

        inputs:

          testRunner: VSTest

          testResultsFiles: "\*\*/tests/\*.trx"

          searchFolder: "$(Common.TestResultsDirectory)"

          failTaskOnFailedTests: true

## 5. Task

Tasks are specified under job/steps/

Different types of tasks built in AzureDevOps, e.g. .netCore, publish test results, bash, docker. Use Azure task template to start with.

Most tasks have inputs and every task can have display name

- task: PublishTestResults@2

        inputs:

          testRunner: VSTest

          testResultsFiles: "\*\*/tests/\*.trx"

          searchFolder: "$(Common.TestResultsDirectory)"

          failTaskOnFailedTests: true

      - task: Docker@2

        displayName: Login to edftradingnonprod

        inputs:

          command: login

          containerRegistry: Compass-DockerNonProdACR

      - task: Bash@3

        displayName: Tag and push images to ACR then cleanup

        inputs:

          targetType: "inline"

          script: |

            #publish to ACR then remove local container and images

            docker image tag compassingestor $(Docker.ContainerRegistry)/compass/ingestor:$(Build.BuildNumber)

            docker image push $(Docker.ContainerRegistry)/compass/ingestor:$(Build.BuildNumber)

            #list containers

            docker ps --no-trunc --all

            #clean up and list images

            echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Clean up local images \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

            docker rmi compassingestor:latest

            docker rmi $(Docker.ContainerRegistry)/compass/ingestor:$(Build.BuildNumber)

            docker image ls

          failOnStderr: true

  - job: CREATE\_ARTIFACTS

    displayName: Publish build artifacts

    pool:

      name: QuantitativeTrading

      demands: ["docker"]

    dependsOn:

      - BUILD\_AND\_TEST\_INGESTOR

    steps:

      - task: Bash@3

        displayName: Add docker tag to images in compose

        inputs:

          filePath: "$(Build.SourcesDirectory)/Code/Release/tag-docker-compose.sh"

          arguments: "$(Build.BuildNumber) docker-compose-ingestor.yml"

          workingDirectory: "$(Build.SourcesDirectory)/Code/Release"

          failOnStderr: true

      - task: CopyFiles@2

        inputs:

          SourceFolder: "$(Build.SourcesDirectory)/Code/Release"

          Contents: "docker-compose-ingestor.yml"

          TargetFolder: "$(Build.ArtifactStagingDirectory)"

      - task: PublishBuildArtifacts@1

        inputs:

          PathtoPublish: "$(Build.ArtifactStagingDirectory)"

          ArtifactName: "compassingestor"

          publishLocation: "Container"

## 6. Publish test results task

This can be part of the build job

- task: PublishTestResults@2

        inputs:

          testRunner: VSTest

          testResultsFiles: "\*\*/tests/\*.trx"

          searchFolder: "$(Common.TestResultsDirectory)"

          failTaskOnFailedTests: true

## 7. Publish build artifacts

In this case, we only publishing the docker compose file because for release we just need to retrieve the docker image from the container registery

  - job: CREATE\_ARTIFACTS

    displayName: Publish build artifacts

    pool:

      name: QuantitativeTrading

      demands: ["docker"]

    dependsOn:

      - BUILD\_AND\_TEST\_INGESTOR

    steps:

      - task: Bash@3

        displayName: Add docker tag to images in compose

        inputs:

          filePath: "$(Build.SourcesDirectory)/Code/Release/tag-docker-compose.sh"

          arguments: "$(Build.BuildNumber) docker-compose-ingestor.yml"

          workingDirectory: "$(Build.SourcesDirectory)/Code/Release"

          failOnStderr: true

      - task: CopyFiles@2

        inputs:

          SourceFolder: "$(Build.SourcesDirectory)/Code/Release"

          Contents: "docker-compose-ingestor.yml"

          TargetFolder: "$(Build.ArtifactStagingDirectory)"

      - task: PublishBuildArtifacts@1

        inputs:

          PathtoPublish: "$(Build.ArtifactStagingDirectory)"

          ArtifactName: "compassingestor"

          publishLocation: "Container"