



.NET DEVOPS REVOLUTION

FLEXIBILITY AND QUALITY CONTROL WITH DOCKER'S ADAPTIVE ARSENAL

WOUTER BAUCHART – SOPRA STERIA

sopra  steria

WHOAMI



Wouter Bauchart

<https://bauchart.be>



.NET Architect @

sopra  steria



.NET, Azure, Azure DevOps, Music, Games, ...



<https://github.com/WouterBau>





<https://github.com/WouterBau/ShadowRoller>

#

**Welcome to #demo-
recording!**

This is the start of the #demo-recording channel.

[Edit Channel](#)



Message #dem...



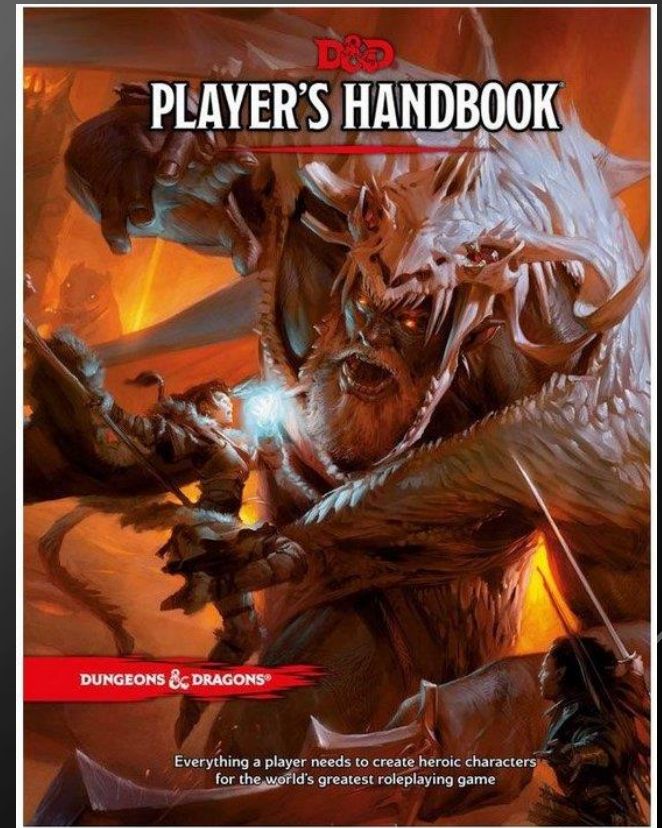
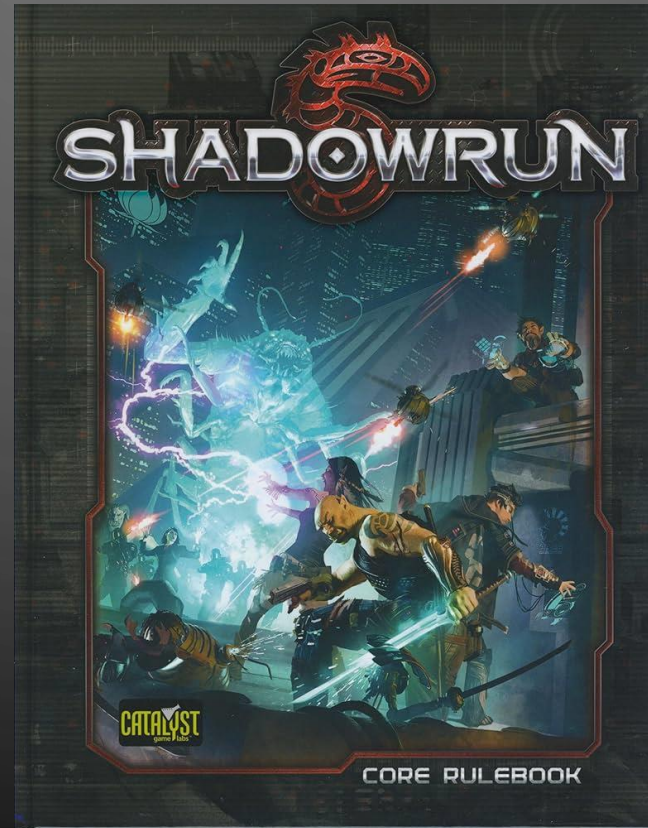
with
CSharpFritz

**Build your first
Discord Chatbot**



Available on LearnTV + .NET Live TV

aka.ms/LearnTV live.dot.net



ISO 9001



README GPL-3.0 license

Azure Pipelines **succeeded**

tests **✓ 41**

coverage **100%**

coverage 86.4%

lines of code 295

technical debt **15min**

vulnerabilities **0**

code smells **5**

bugs **0**

Mutation score **100%**

ShadowRoller

A small Discord bot currently set up as an easier way to resolve ShadownRun 5 tests. It reads and sends messages in the channel they're invited to. It calculates the hits and whether you (critically) glitched.



Unit Tests



Code Coverage

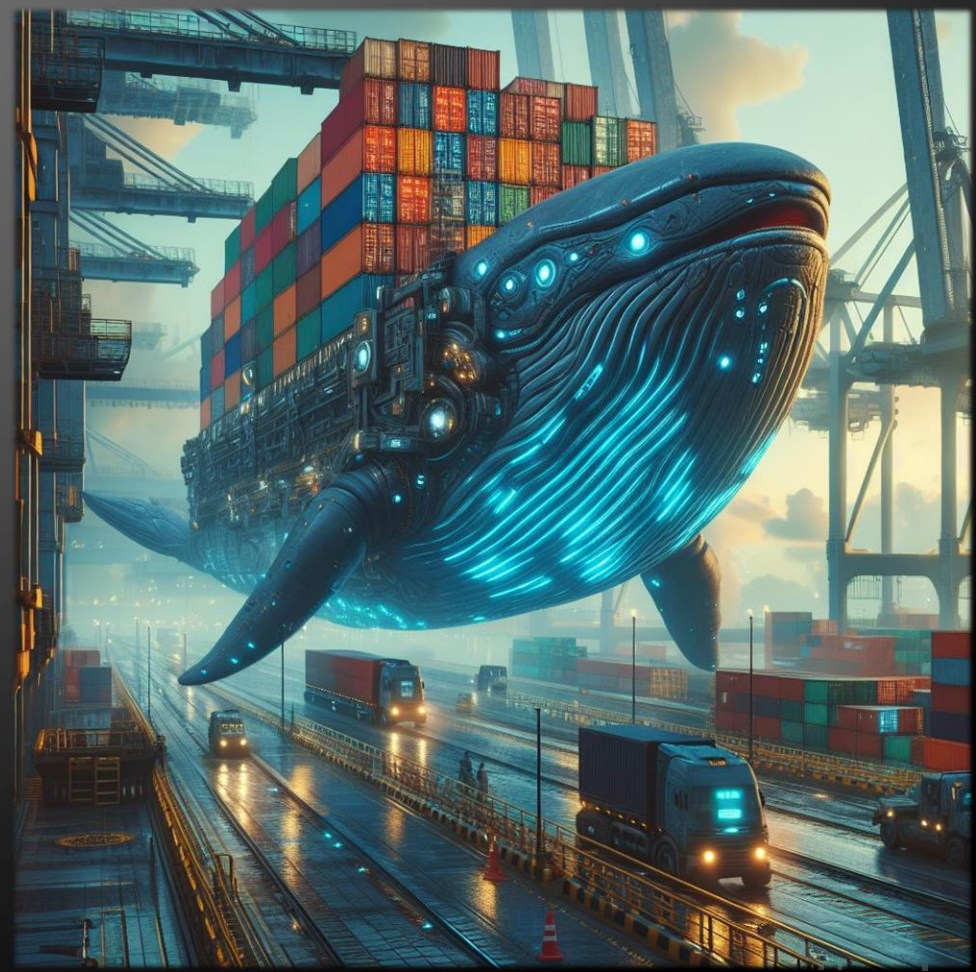


Static Code Analysis
SonarCloud

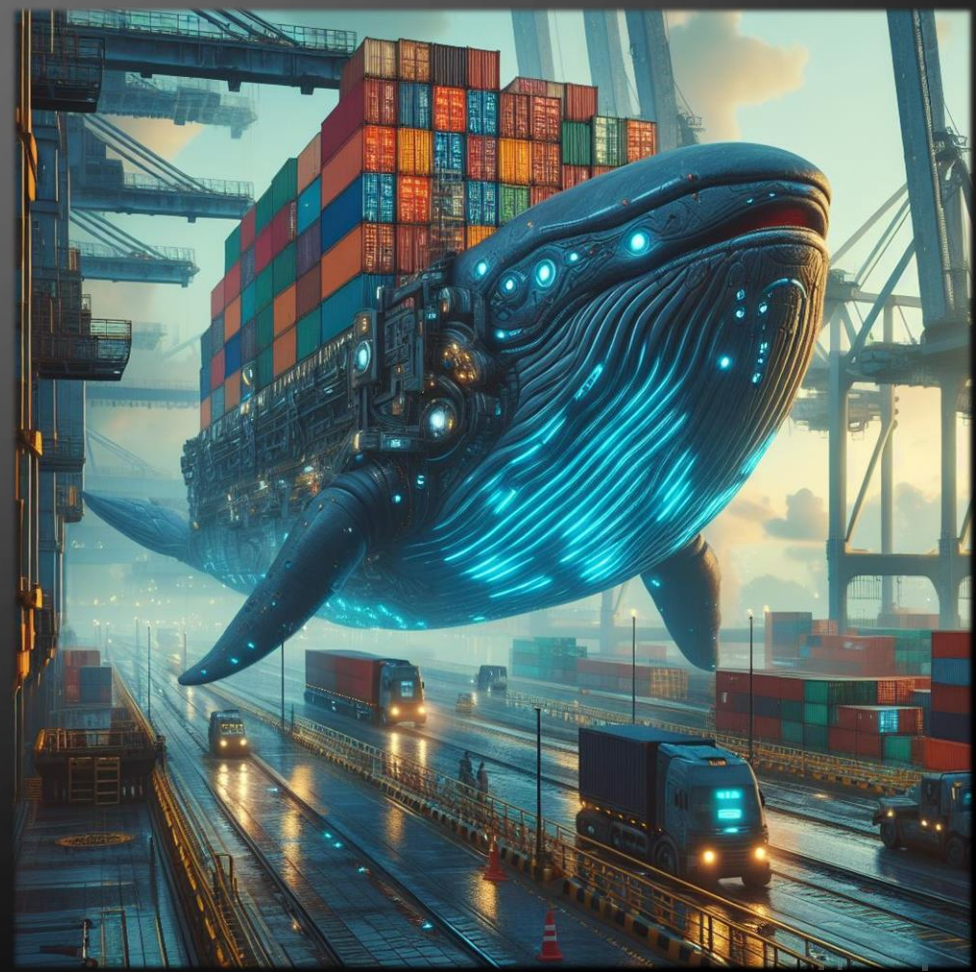


Mutation Testing
Stryker.NET

WHAT RUNS?



WHAT RUNS?



EPIPHANY SOURCES



- Azure DevOps Documentation: PublishTestResults *
 - (Tedious?) Docker build, run, cp and stop sample
- .NET Docker samples: Complex App *
 - (Smooth) Docker build and run with volume binding sample
- Party of DevOps colleagues + Hands-On experience



* Links at the end of the show

WHAT DO WE GAIN IN OUR TOOLING ARSENAL?



- Reproducibility & Consistency
 - Dev Machine == Build Agent executions
 - Reproducible (integration) test environments
- Stability
 - Smaller dependency on build agents
 - Expected tooling versions used
- Flexibility
 - Unrestricted in how to create your build
 - Unrestricted on where to create your build
 - Unrestricted on what CI/CD platform you use



TAKE THESE INTO ACCOUNT



- Build agents
 - Require support for containers
- Build & Run Command
 - Ensure all expected arguments are provided




OPTIONAL TASKS FOR CI/CD PLATFORM

THESE COULD ALSO BE PART OF YOUR DOCKERFILE




- Pipelines
 - Store results for run



Summary Tests Code Coverage Mutation Report

Manually run by  Wouter Bauchart


[View change](#)


Repository and version

 WouterBau/ShadowRoller


 demo/devops-revo-4-rec  d26595a7


Time started and elapsed

 Apr 27 at 12:54 AM


 2m 17s


Related

 0 work items

 0 artifacts

Tests and coverage

 100% passed

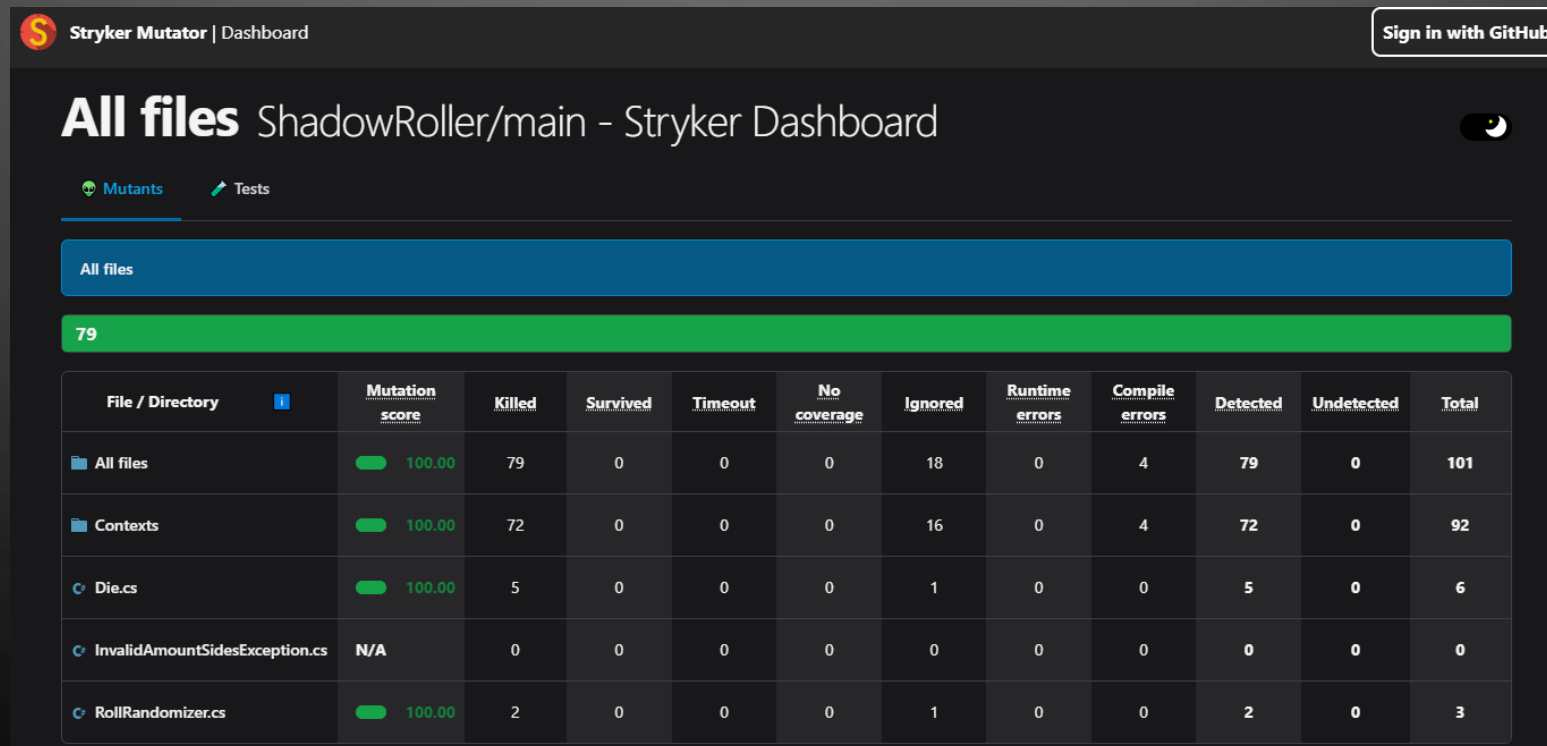
 100.00% covered

OPTIONAL TASKS FOR CI/CD PLATFORM

THESE COULD ALSO BE PART OF YOUR DOCKERFILE



- Dashboards
 - Visualize the results



SHOWTIME : CODE COVERAGE



<https://github.com/WouterBau/ShadowRoller-NET-DevOps-Revolution-Docker/tree/demo/devops-revo-1>



You, last week | 1 author (You)

```
# https://hub.docker.com/_/microsoft-dotnet
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
WORKDIR /source
```

```
# Copy each .csproj file individually
COPY src/ShadowRoller.Domain/ShadowRoller.Domain.csproj src/ShadowRoller.
Domain/
COPY src/ShadowRoller.DiscordBot/ShadowRoller.DiscordBot.csproj src/
ShadowRoller.DiscordBot/
COPY tests/ShadowRoller.Domain.Tests/ShadowRoller.Domain.Tests.csproj tests/
ShadowRoller.Domain.Tests/
```

```
# Copy the solution file
COPY ShadowRoller.sln ./

# Restore the projects in the solution
RUN dotnet restore ShadowRoller.sln
```

```
# Copy the rest of the application code
COPY . .
```

```
# Build the application
RUN dotnet build --no-restore ShadowRoller.sln
```

```
# Run the tests
FROM build AS test
ENTRYPOINT ["dotnet", "test", "ShadowRoller.sln", "--no-build", "/
p:CollectCoverage=true", "/
p:CoverletOutputFormat=json%2ccobertura%2ccopencover", "/p:CoverletOutput=../
TestResults/", "/p:MergeWith=../TestResults/coverage.json", "/
p:SkipAutoProps=true", "/m:1"]
```



SHOWTIME : CODE COVERAGE



```
docker build --pull --rm -t shadowroller:qa .
```

```
docker run -rm \  
-v ${pwd}/TestResults:/source/tests/TestResults \  
shadowroller:qa
```

<https://github.com/WouterBau/ShadowRoller-NET-DevOps-Revolution-Docker/tree/demo/devops-revo-1>

SHOWTIME : MUTATION TESTING



<https://github.com/WouterBau/ShadowRoller-NET-DevOps-Revolution-Docker/tree/demo/devops-revo-2>

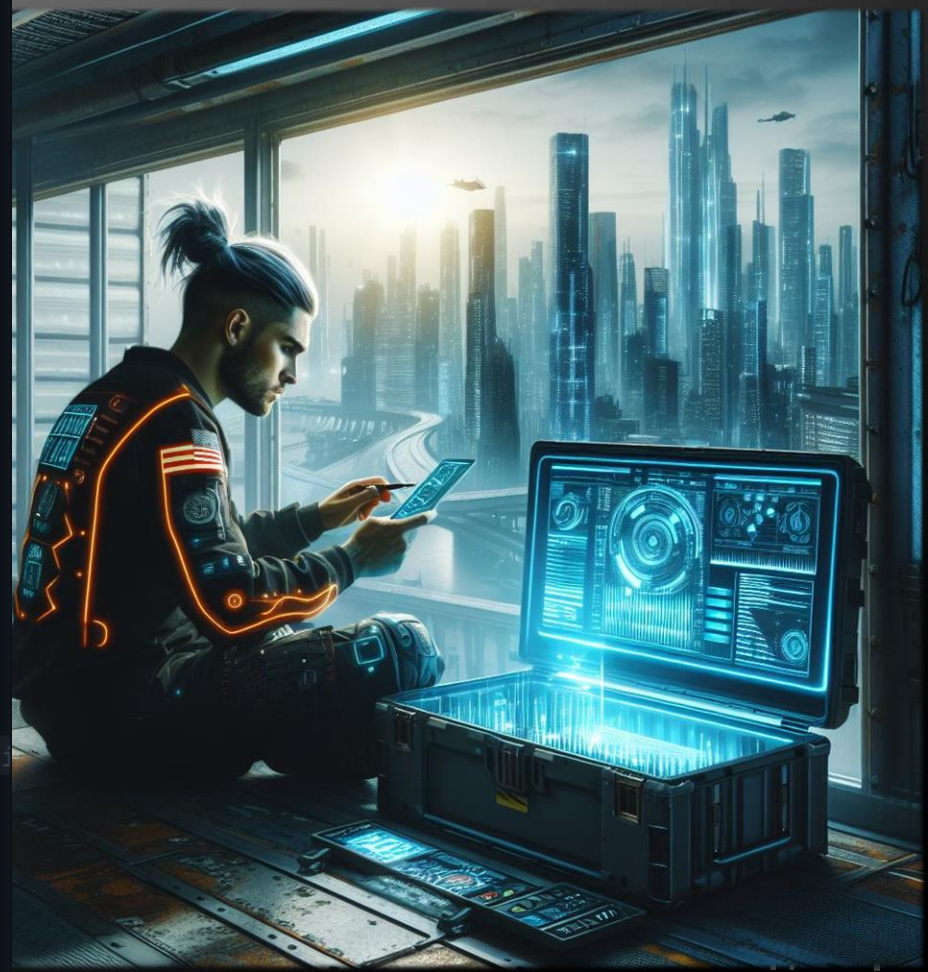


```
1 # https://hub.docker.com/_/microsoft-dotnet
2 FROM mcr.microsoft.com/dotnet/sdk:8.0 AS copyrestore
3 WORKDIR /source
4 # Copy each .csproj file individually
5 COPY src/ShadowRoller.Domain/ShadowRoller.Domain.csproj src/ShadowRoller.Domain/
6 COPY src/ShadowRoller.DiscordBot/ShadowRoller.DiscordBot.csproj src/ShadowRoller.
  DiscordBot/
7 COPY tests/ShadowRoller.Domain.Tests/ShadowRoller.Domain.Tests.csproj tests/ShadowRoller.
  Domain.Tests/
8 # Copy the solution file
9 COPY ShadowRoller.sln ./
10 # Restore the projects in the solution
11 RUN dotnet restore ShadowRoller.sln
12 # Copy the rest of the application code
13 COPY . .
```

```
15 FROM copyrestore AS build
16 # Build the application
17 RUN dotnet build --no-restore ShadowRoller.sln
18
19 # Run the tests
20 FROM build AS test
21 ENTRYPOINT ["dotnet", "test", "ShadowRoller.sln", "--no-build", "/
  p:CollectCoverage=true", "/p:CoverletOutputFormat=json%2ccobertura%2copencover", "/
  p:CoverletOutput=../TestResults/", "/p:MergeWith=../TestResults/coverage.json", "/
  p:SkipAutoProps=true", "/m:1"]
```

```
23 FROM copyrestore as mutation
24 # Install dotnet-stryker
25 RUN dotnet tool install --global dotnet-stryker
26 # Add dotnet tools to PATH
27 ENV PATH="$PATH:/root/.dotnet/tools"
28 ENTRYPOINT dotnet stryker -s ShadowRoller.sln --output MutationResults --dashboard-api-key
  $STRYKER_API_KEY -v $BRANCH_NAME
```

You, 7 days ago • feat: Provided Mutac



SHOWTIME : MUTATION TESTING



```
docker build --pull -rm \  
--target mutation -t shadowroller:qa-mutation .
```

```
docker run --rm --env-file .env \  
-e BRANCH_NAME=$BRANCH \  
-v ${pwd}/MutationResults:/source/MutationResults \  
shadowroller:qa-mutation
```

<https://github.com/WouterBau/ShadowRoller-NET-DevOps-Revolution-Docker/tree/demo/devops-revo-2>

SHOWTIME : SONARCLOUD



<https://github.com/WouterBau/ShadowRoller-NET-DevOps-Revolution-Docker/tree/demo/devops-revo-3>





Dockerfile > ...

You, 4 days ago | 1 author (You)

```
1 # https://hub.docker.com/_/microsoft-dotnet
2 FROM mcr.microsoft.com/dotnet/sdk:8.0 AS copyrestore
3 WORKDIR /source
4 # Copy each .csproj file individually
5 COPY src/ShadowRoller.Domain/ShadowRoller.Domain.csproj src/ShadowRoller.Domain/
6 COPY src/ShadowRoller.DiscordBot/ShadowRoller.DiscordBot.csproj src/ShadowRoller.
  DiscordBot/
7 COPY tests/ShadowRoller.Domain.Tests/ShadowRoller.Domain.Tests.csproj tests/
  ShadowRoller.Domain.Tests/
8 # Copy the solution file
9 COPY ShadowRoller.sln ./
10 # Restore the projects in the solution
11 RUN dotnet restore ShadowRoller.sln
12 # Copy the rest of the application code
13 COPY . .
14
15 FROM copyrestore as mutation
16 # Install dotnet-stryker
17 RUN dotnet tool install --global dotnet-stryker
18 # Add dotnet tools to PATH
19 ENV PATH="$PATH:/root/.dotnet/tools"
20 ENTRYPOINT dotnet stryker -s ShadowRoller.sln --output MutationResults
  --dashboard-api-key $STRYKER_API_KEY -v $BRANCH_NAME
```

```
22 FROM copyrestore as qabuild
23 ARG SONAR_HOST_URL
24 ARG SONAR_TOKEN
25 ARG SONAR_ORGANIZATION
26 ARG SONAR_PROJECT_KEY
27 ARG SONAR_PROJECT_NAME
28 ARG SONAR_BRANCH
29
30 # Install OpenJDK 17
31 RUN apt-get update && apt-get install -y openjdk-17-jdk
32 # Install sonarqube scanner
33 RUN dotnet tool install --global dotnet-sonarscanner
34 # Add dotnet tools to PATH
35 ENV PATH="$PATH:/root/.dotnet/tools"
36 # Begin SonarCloud analysis
```

Dockerfile >

```
34 # Add dotnet tools to PATH
35 ENV PATH="$PATH:/root/.dotnet/tools"
36 # Begin SonarCloud analysis
37 RUN dotnet sonarscanner begin \
38   /o:"$SONAR_ORGANIZATION" \
39   /k:"$SONAR_PROJECT_KEY" \
40   /n:"$SONAR_PROJECT_NAME" \
41   /d:sonar.token="$SONAR_TOKEN" \
42   /d:sonar.host.url="$SONAR_HOST_URL" \
43   /d:sonar.branch.name="$SONAR_BRANCH" \
44   /d:sonar.cs.opencover.reportsPaths="**/coverage.opencover.xml"
45 # Rebuild the application
46 RUN dotnet build --no-restore ShadowRoller.sln
47 # Run the tests
48 RUN dotnet test ShadowRoller.sln --no-build /p:CollectCoverage=true /
  p:CoverletOutputFormat=json%2ccobertura%2copencover /p:CoverletOutput=../
  TestResults/ /p:MergeWith=../TestResults/coverage.json /p:SkipAutoProps=true /m:1
49 # End SonarCloud analysis
50 RUN dotnet sonarscanner end /d:sonar.token="$SONAR_TOKEN"
51
52 FROM qabuild as qatest
53 ENTRYPOINT [ "cp", "-r", "./tests/TestResults", "./volume/TestResults" ]
54
55 FROM copyrestore AS build
56 # Build the application
57 RUN dotnet build --no-restore ShadowRoller.sln
```



SHOWTIME : MUTATION TESTING

```
docker build --pull -rm \  
  --build-arg SONAR_HOST_URL=$SONAR_HOST \  
  --build-arg SONAR_TOKEN=$SONAR_TOKEN \  
  --build-arg SONAR_ORGANIZATION=$SONAR_ORGANIZATION \  
  --build-arg SONAR_PROJECT_KEY=$SONAR_PROJECT_KEY \  
  --build-arg SONAR_PROJECT_NAME=$SONAR_PROJECT_NAME \  
  --build-arg SONAR_BRANCH=$BRANCH \  
  --target qatest -t shadowroller:qa-tests .
```

```
docker run --rm -v ${pwd}/TestResults:/source/volume/TestResults \  
shadowroller:qa-tests
```

<https://github.com/WouterBau/ShadowRoller-NET-DevOps-Revolution-Docker/tree/demo/devops-revo-3>

SHOWTIME : AZURE DEVOPS PIPELINE



<https://github.com/WouterBau/ShadowRoller-NET-DevOps-Revolution-Docker/tree/demo/devops-revo-4>





azure-pipelines > qa-azure-pipelines.yml > jobs > 0 > steps > 0 > script

You, 1 hour ago | 1 author (You)

trigger: none

variables:

- group: stryker-variable-group
- group: sonarcloud-variable-group
- name: SONAR_PROJECT_KEY
- value: WouterBau_ShadowRoller
- name: SONAR_PROJECT_NAME
- value: ShadowRoller

pool:

vmImage: 'ubuntu-latest'

jobs:

- job: sast

steps:

- script: |

```
docker build --pull --rm --build-arg SONAR_HOST_URL=$(SONAR_HOST_URL)
--build-arg SONAR_TOKEN=$(SONAR_TOKEN) --build-arg SONAR_ORGANIZATION=$(
SONAR_ORGANIZATION) --build-arg SONAR_PROJECT_KEY=$(SONAR_PROJECT_KEY)
--build-arg SONAR_PROJECT_NAME=$(SONAR_PROJECT_NAME) --build-arg
SONAR_BRANCH=$(Build.SourceBranchName) --target qatest -t
shadowroller:qa-tests .
docker run --rm -v $(System.DefaultWorkingDirectory)/TestResults-Docker:/
source/volume/TestResults shadowroller:qa-tests
```

- task: PublishTestResults@2

azure-pipelines > qa-azure-pipelines.yml > jobs > 0 > steps

14 jobs:

- job: sast

16 steps:

- task: PublishTestResults@2

inputs:

- testRunner: VSTest
- testResultsFiles: '\$(System.DefaultWorkingDirectory)/**/*.trx'
- failTaskOnFailedTests: true

- task: PublishCodeCoverageResults@2

inputs:

- codeCoverageTool: 'Cobertura'
- summaryFileLocation: '\$(System.DefaultWorkingDirectory)/**/coverage.cobertura.xml'

- job: mutation

steps:

- script: |

```
docker build --pull --rm --target mutation -t shadowroller:qa-mutation .
docker run --rm -e STRYKER_API_KEY=$(STRYKER_API_KEY) -e BRANCH_NAME=$(Build
SourceBranchName) -v $(System.DefaultWorkingDirectory)/MutationResults:/
source/MutationResults shadowroller:qa-mutation
```

- task: PublishMutationReport@1

displayName: 'Publish Mutation Test Report'

inputs:

- reportPattern: '\$(System.DefaultWorkingDirectory)/**/mutation-report.html'

47 # Run the tests

48 RUN dotnet test ShadowRoller.sln --no-build --logger trx --results-directory ./TestResults/ /
p:CollectCoverage=true /p:CoverletOutputFormat=json%2ccobertura%2ccopencover /
p:CoverletOutput=../TestResults/ /p:MergeWith=../TestResults/coverage.json /
p:SkipAutoProps=true /m:1

You, 1 hour ago • Merge branch 'temp/test-dockerized-azdevops'

Report does not exist

WHERE TO GO NEXT?



- Publish hybrid/desktop apps?
- How to handle NuGet packages?
- Moving to another CI/CD platform?



REFERENCES



- Azure DevOps 'PublishTestResults' Documentation:
<https://learn.microsoft.com/en-us/azure/devops/pipelines/tasks/reference/publish-test-results-v2>
- .NET Docker samples complexapp:
<https://github.com/dotnet/dotnet-docker/tree/main/samples/complexapp>
- More reference information:
<https://bauchart.be/2024/05/25/net-devops-revolution/>



<https://github.com/WouterBau/ShadowRoller-NET-DevOps-Revolution-Docker>