.NET DEVOPS REVOLUTION

FLEXIBILITY AND QUALITY CONTROL WITH DOCKER'S ADAPTIVE ARSENAL

WOUTER BAUCHART – SOPRA STERIA

WHOAMI

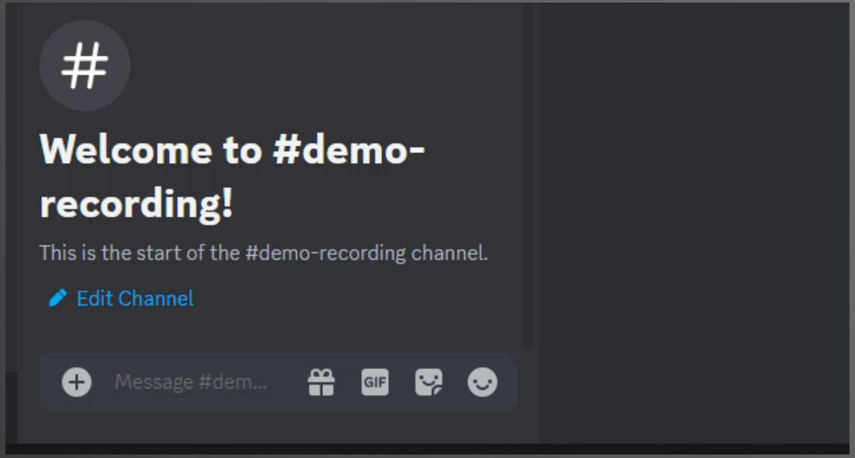
Wouter Bauchart
https://bauchart.be

.NET Architect @ sopra steria

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

https://github.com/WouterBau

SHADOWROLLER



#

Welcome to #demo-recording!

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

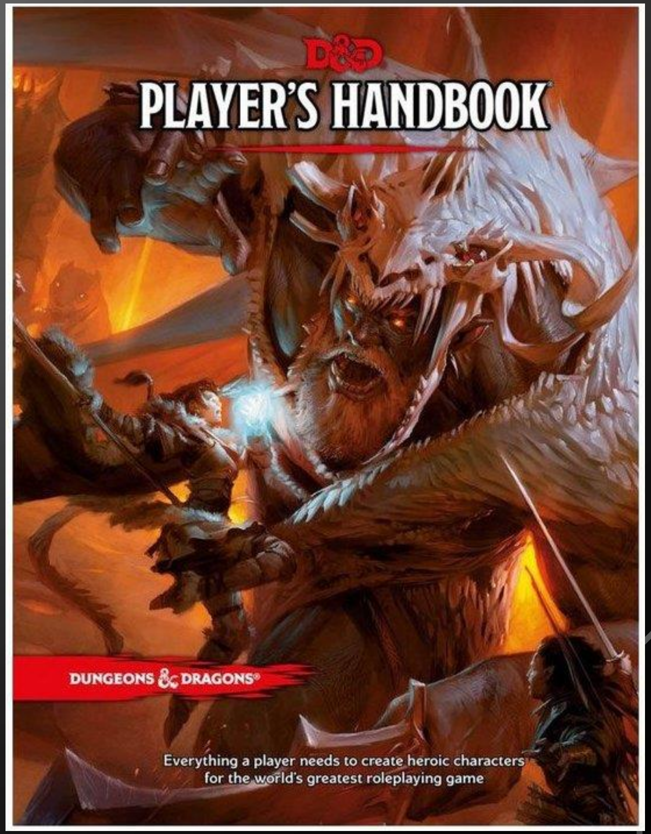
https://github.com/WouterBau/ShadowRoller

with CSharpFritz

Build your first Discord Chatbot

Available on LearnTV + .NET Live TV

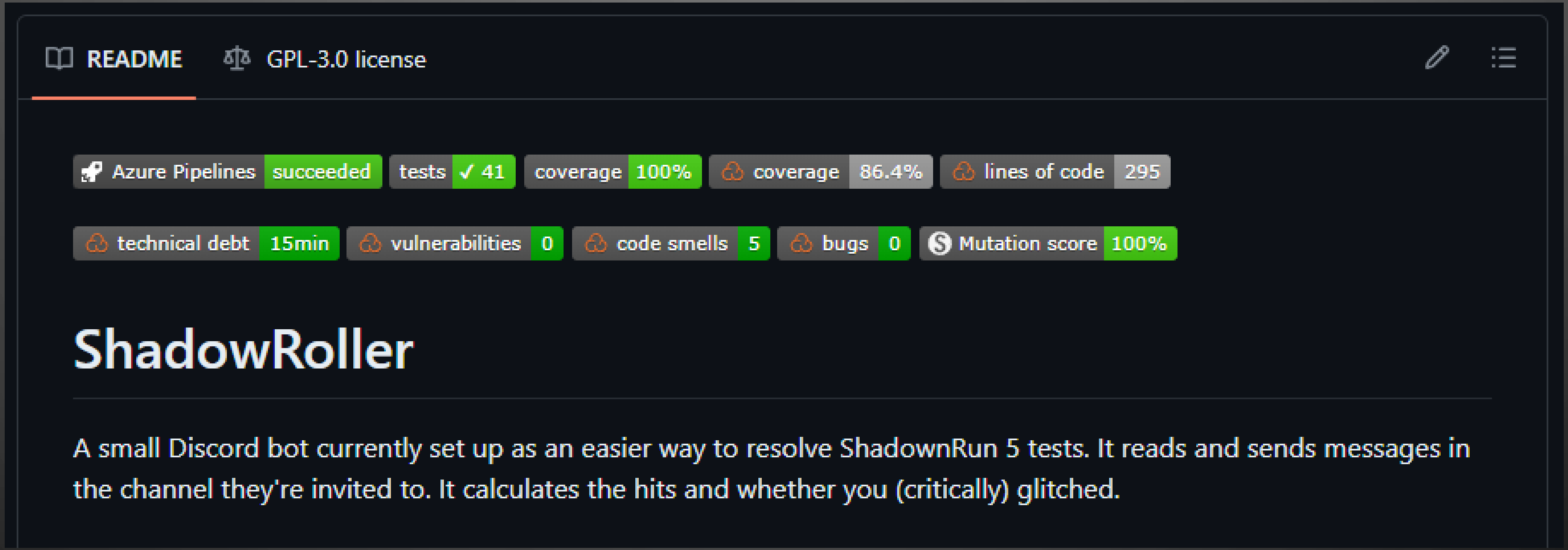
aka.ms/LearnTV live.dot.net



D&D
PLAYER'S HANDBOOK
DUNGEONS & DRAGONS
Everything a player needs to create heroic characters for the world's greatest roleplaying game

ISO 9001 ✓

SHADOWROLLER



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.

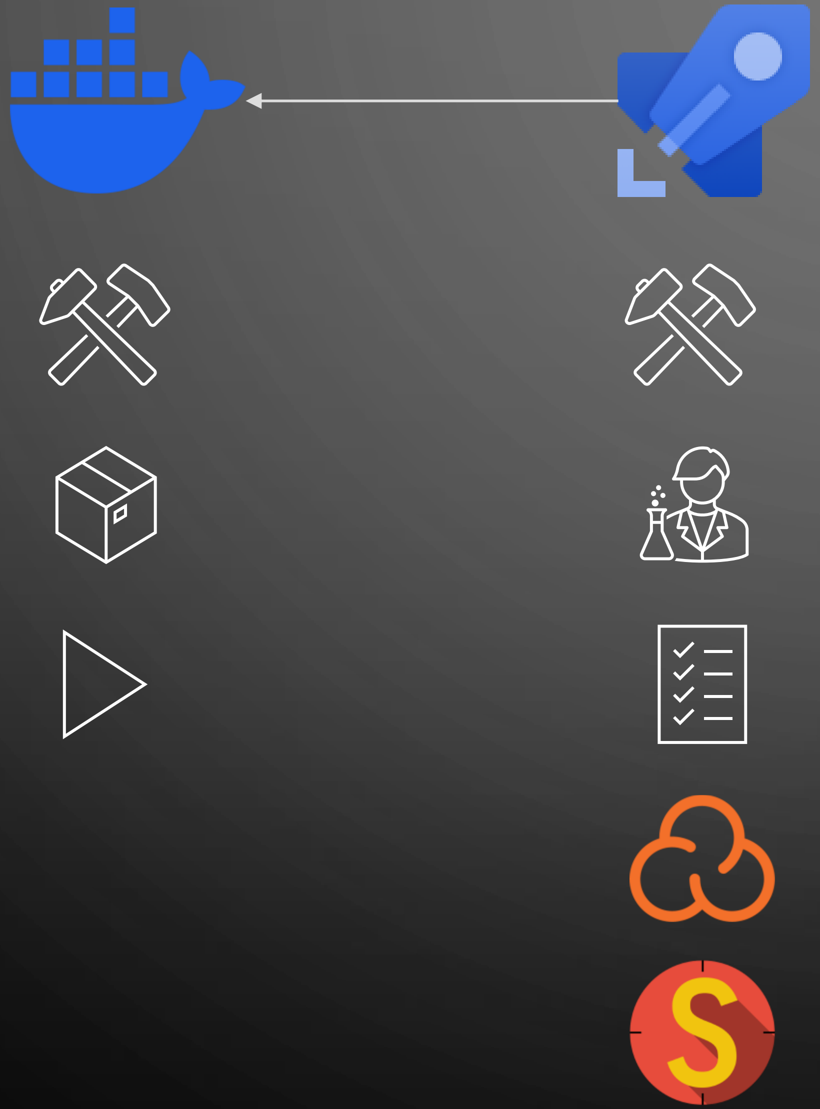
Unit Tests

Code Coverage

Static Code Analysis
SonarCloud

Mutation Testing
Stryker.NET

WHAT RUNS?



SHADOWROLLER

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

SHADOWROLLER

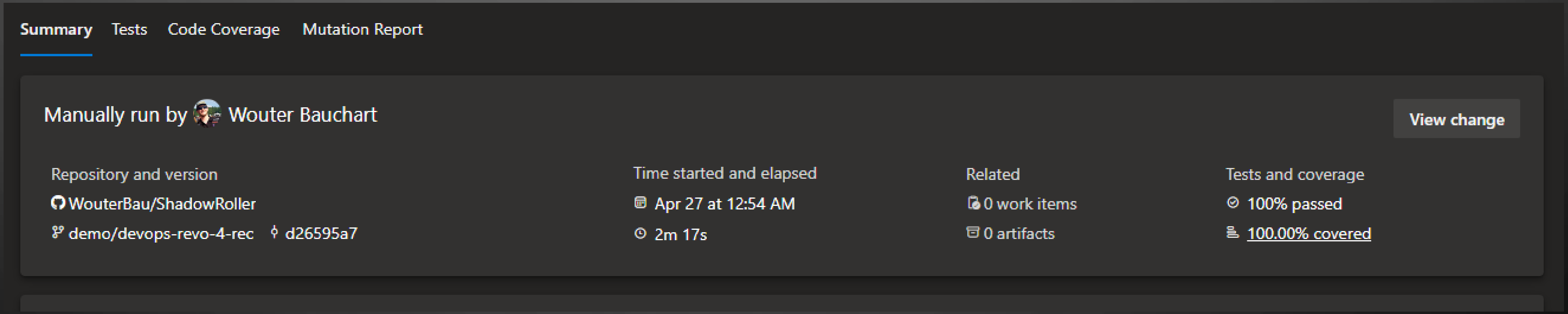
TAKE THESE INTO ACCOUNT

- 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



Manually run by Wouter Bauchart

Repository and version
WouterBau/ShadowRoller
demo/devops-revo-4-rec d26595a7

Time started and elapsed
Apr 27 at 12:54 AM
2m 17s

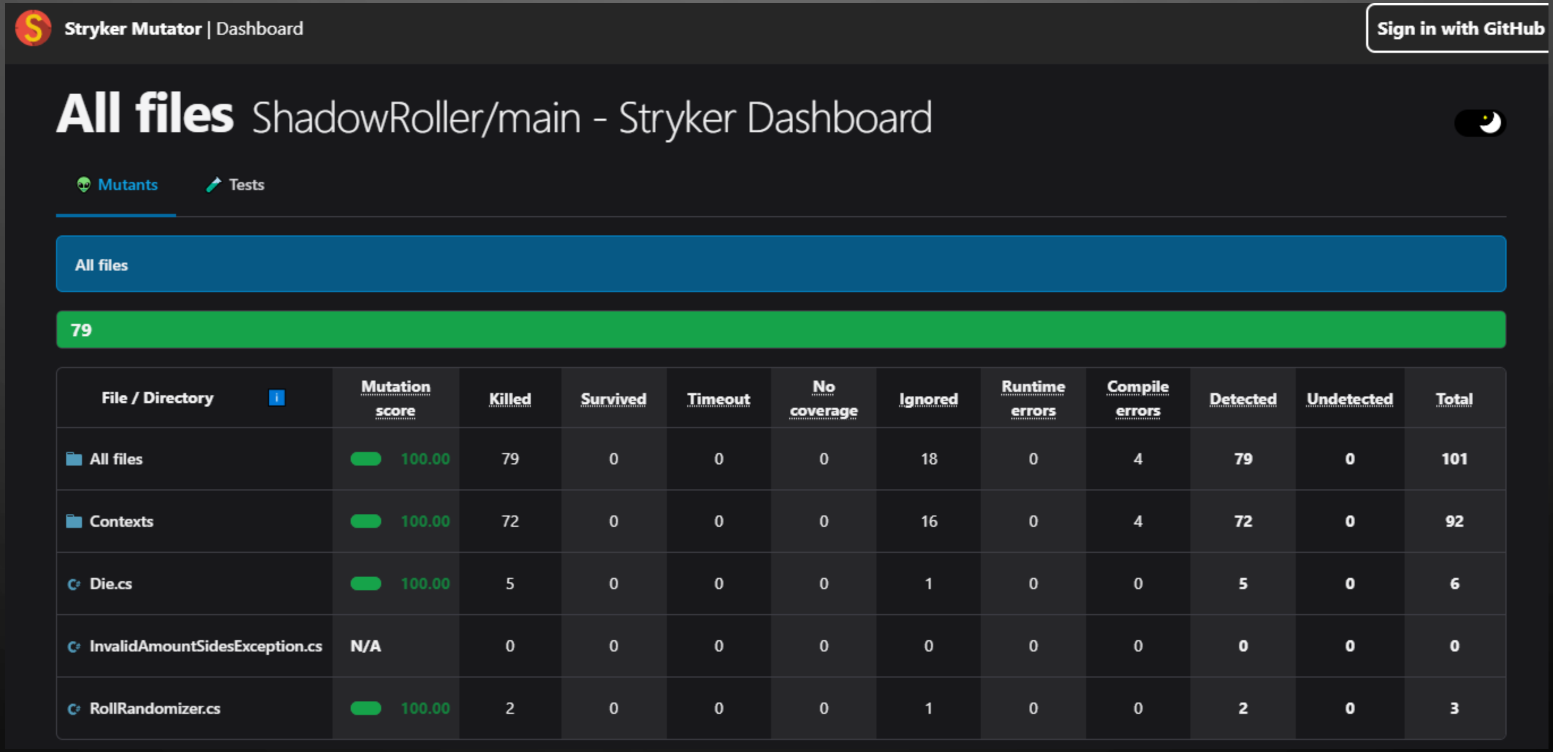
Related
0 work items
0 artifacts

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



All files ShadowRoller/main - Stryker Dashboard

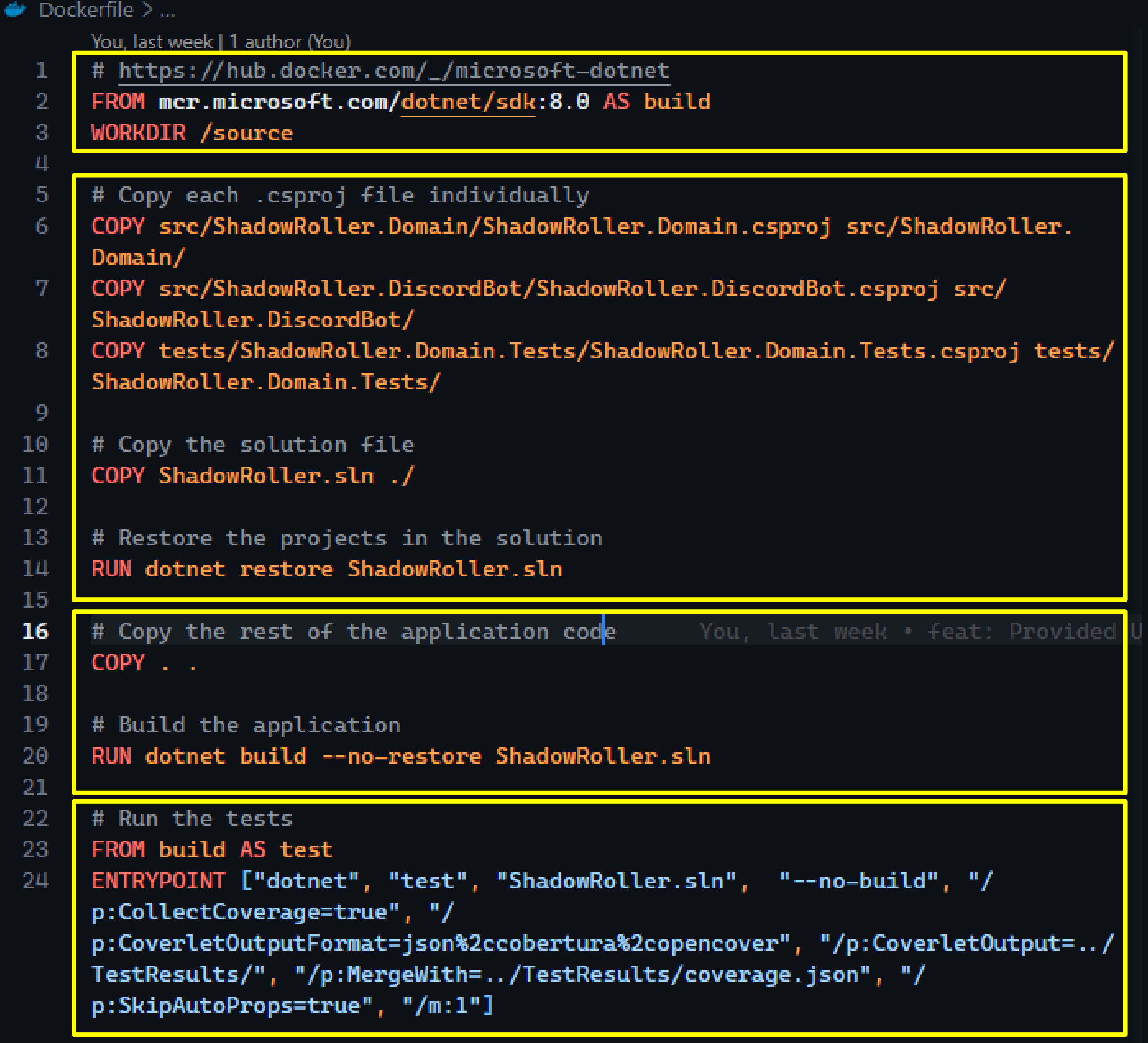
All files

79

File / Directory	Mutation score	Killed	Survived	Timeout	No coverage	Ignored	Runtime errors	Compile errors	Detected	Undetected	Total
All files	100.00	79	0	0	0	18	0	4	79	0	101
Contexts	100.00	72	0	0	0	16	0	4	72	0	92
Die.cs	100.00	5	0	0	0	1	0	0	5	0	6
InvalidAmountSidesException.cs	N/A	0	0	0	0	0	0	0	0	0	0
RollRandomizer.cs	100.00	2	0	0	0	1	0	0	2	0	3

SHOWTIME : CODE COVERAGE

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


Dockerfile > ...
You, last week | 1 author (You)
You, last week • feat: Provided U

```dockerfile
# 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%2copencover", "/p:CoverletOutput=../TestResults/", "/p:MergeWith=../TestResults/coverage.json", "/p:SkipAutoProps=true", "/m:1"]
```

SHADOWROLLER

SHOWTIME : CODE COVERAGE

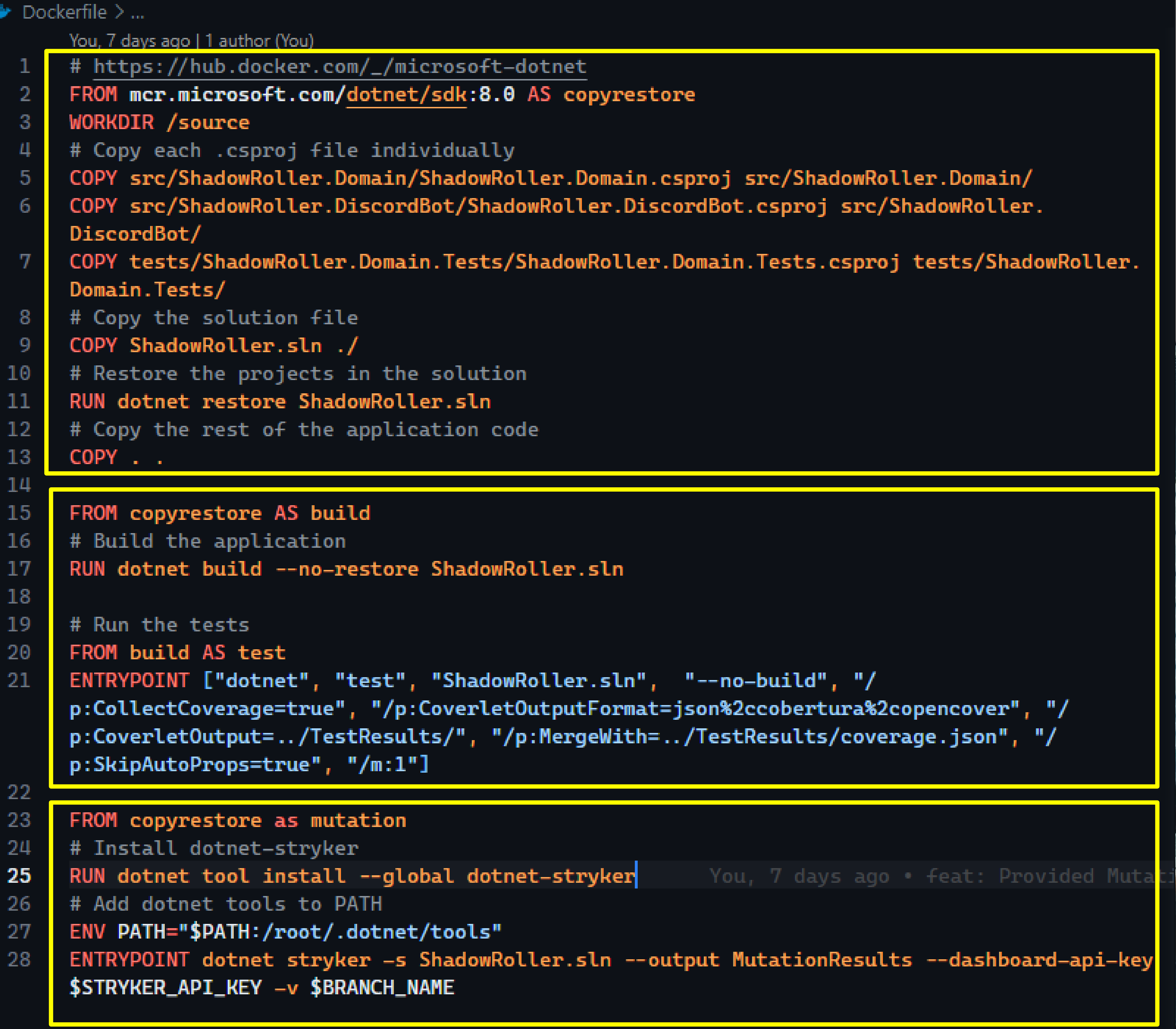
```
docker build --pull --rm -t shadowroller:qa .
```

```
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



Dockerfile > ...

You, 7 days ago | 1 author (You)

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

FROM copyrestore AS build
# 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%2copencover", "/p:CoverletOutput=../TestResults/", "/p:MergeWith=../TestResults/coverage.json", "/p:SkipAutoProps=true", "/m:1"]

FROM copyrestore as mutation
# Install dotnet-stryker
RUN dotnet tool install --global dotnet-stryker        You, 7 days ago • feat: Provided Muta
# Add dotnet tools to PATH
ENV PATH="$PATH:/root/.dotnet/tools"
ENTRYPOINT dotnet stryker -s ShadowRoller.sln --output MutationResults --dashboard-api-key $STRYKER_API_KEY -v $BRANCH_NAME
```

SHADOWROLLER

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
```

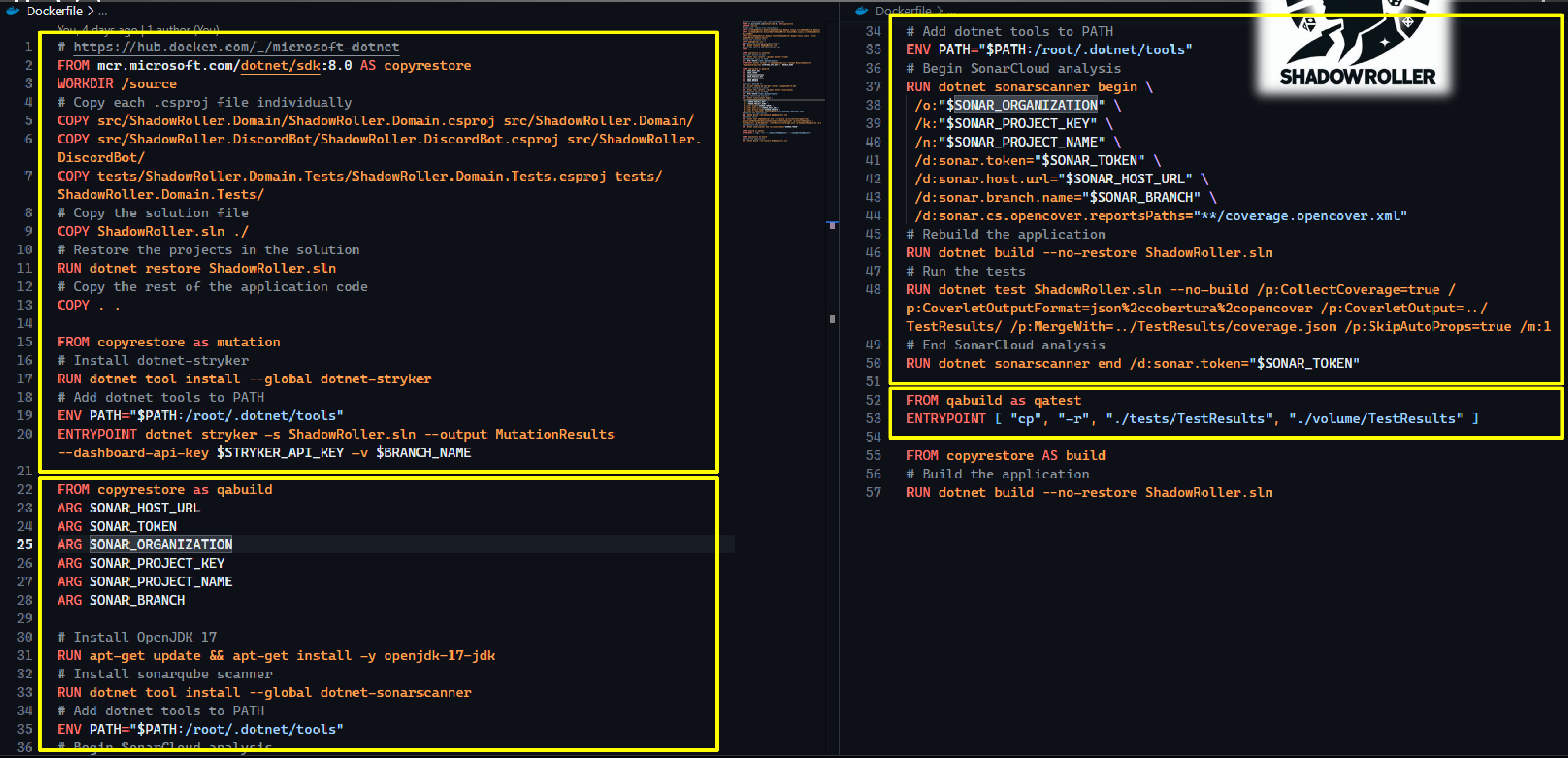
https://github.com/WouterBau/ShadowRoller-NET-DevOps-Revolution-Docker/tree/demo/devops-revo-2

SHOWTIME : SONARCLOUD

SHADOWROLLER

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

SHADOWROLLER



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

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

FROM copyrestore as qabuild
ARG SONAR_HOST_URL
ARG SONAR_TOKEN
ARG SONAR_ORGANIZATION
ARG SONAR_PROJECT_KEY
ARG SONAR_PROJECT_NAME
ARG SONAR_BRANCH

# Install OpenJDK 17
RUN apt-get update && apt-get install -y openjdk-17-jdk
# Install sonarqube scanner
RUN dotnet tool install --global dotnet-sonarscanner
# Add dotnet tools to PATH
ENV PATH="$PATH:/root/.dotnet/tools"
# Begin SonarCloud analysis
RUN dotnet sonarscanner begin \
 /o:"$SONAR_ORGANIZATION" \
 /k:"$SONAR_PROJECT_KEY" \
 /n:"$SONAR_PROJECT_NAME" \
 /d:sonar.token="$SONAR_TOKEN" \
 /d:sonar.host.url="$SONAR_HOST_URL" \
 /d:sonar.branch.name="$SONAR_BRANCH" \
 /d:sonar.cs.opencover.reportsPaths="**/coverage.opencover.xml"
# Rebuild the application
RUN dotnet build --no-restore ShadowRoller.sln
# Run the tests
RUN dotnet test ShadowRoller.sln --no-build /p:CollectCoverage=true /p:CoverletOutputFormat=json%2ccobertura%2copencover /p:CoverletOutput=../TestResults/ /p:MergeWith=../TestResults/coverage.json /p:SkipAutoProps=true /m:1
# End SonarCloud analysis
RUN dotnet sonarscanner end /d:sonar.token="$SONAR_TOKEN"

FROM qabuild as qatest
ENTRYPOINT [ "cp", "-r", "./tests/TestResults", "./volume/TestResults" ]

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

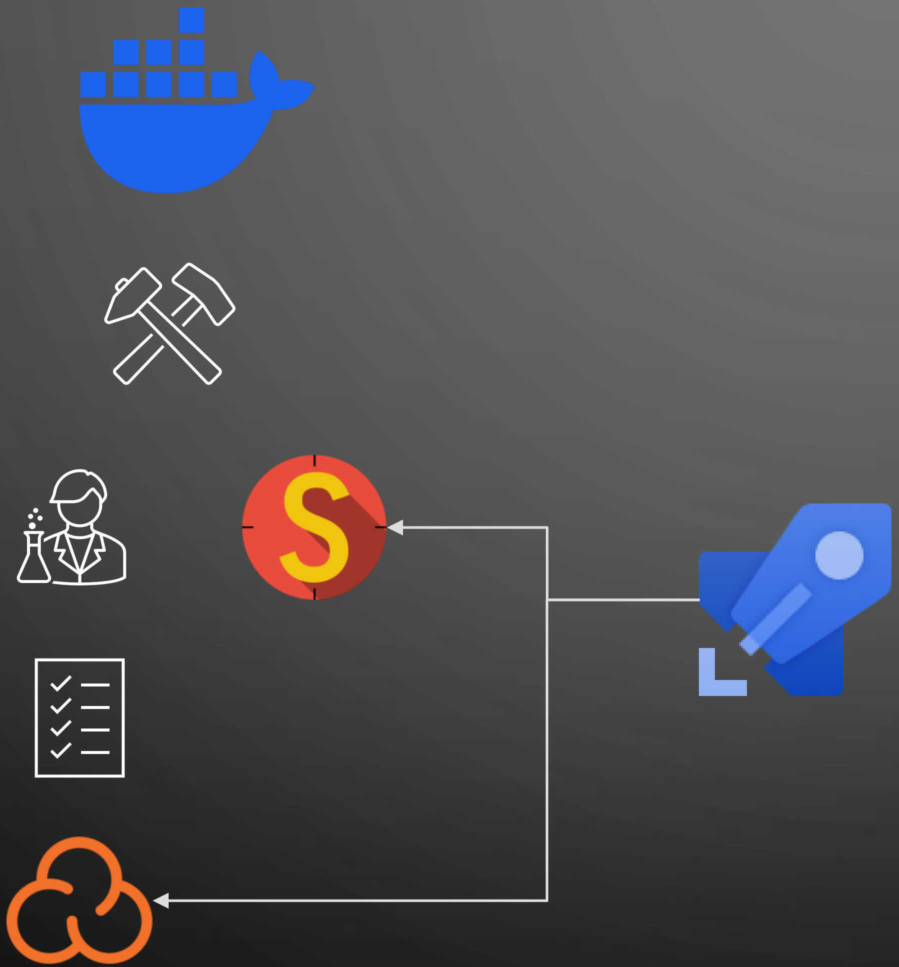
SHOWTIME : MUTATION TESTING

```
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 .
```

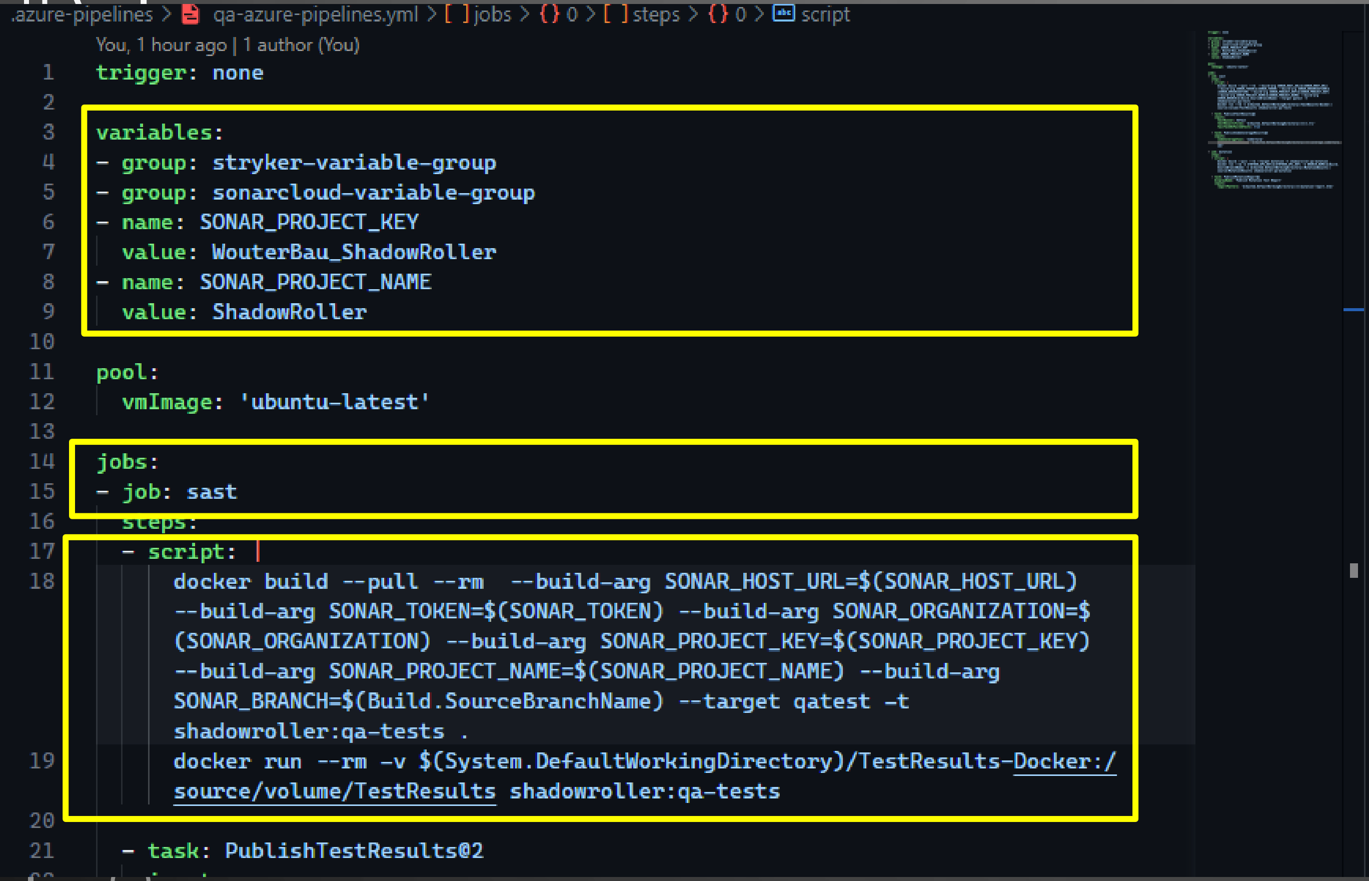
```
docker run --rm -v ${pwd}/TestResults:/source/volume/TestResults \
shadowroller:qa-tests
```

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

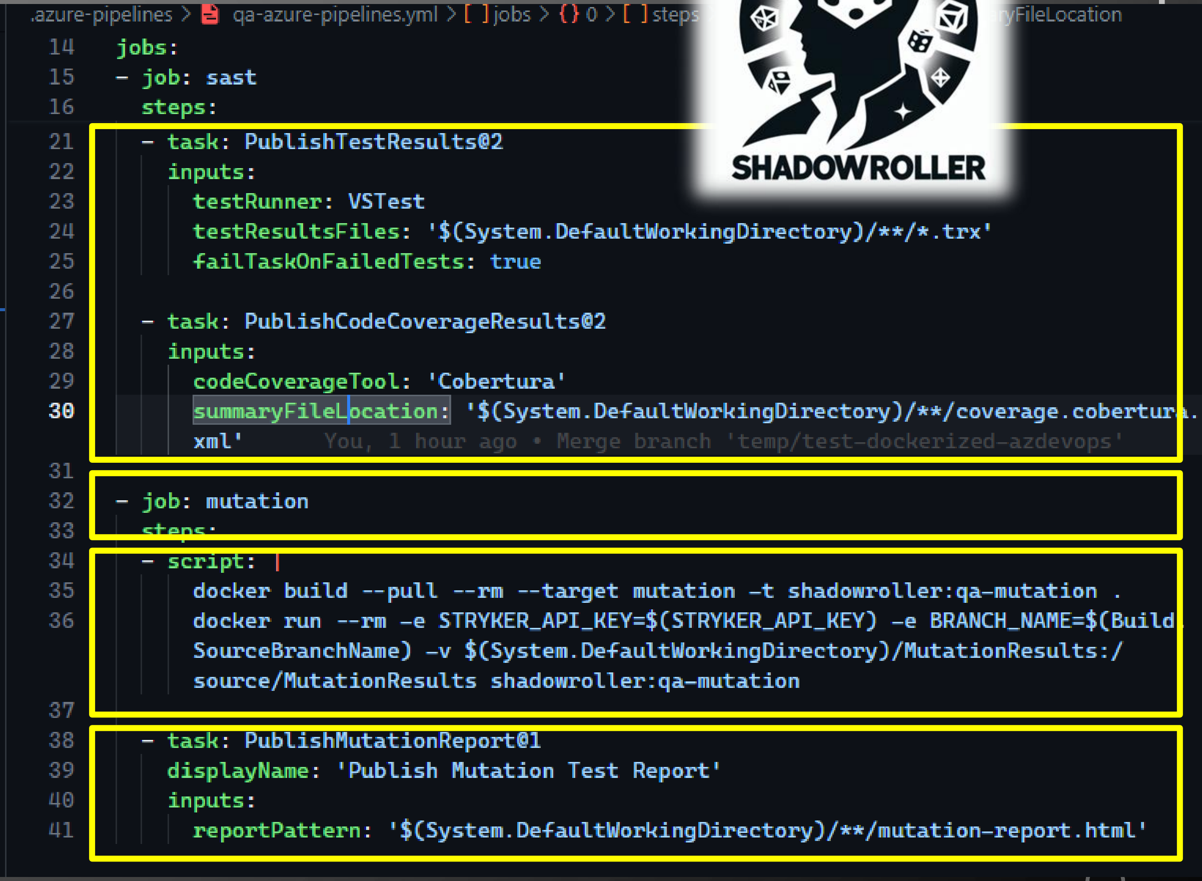
SHOWTIME : AZURE DEVOPS PIPELINE



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

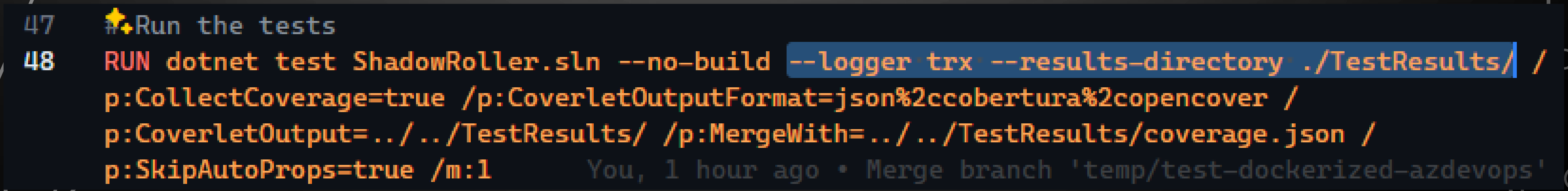

.azure-pipelines > qa-azure-pipelines.yml > [ ] jobs > {} 0 > [ ] steps > {} 0 > script
You, 1 hour ago | 1 author (You)

```yaml
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
```

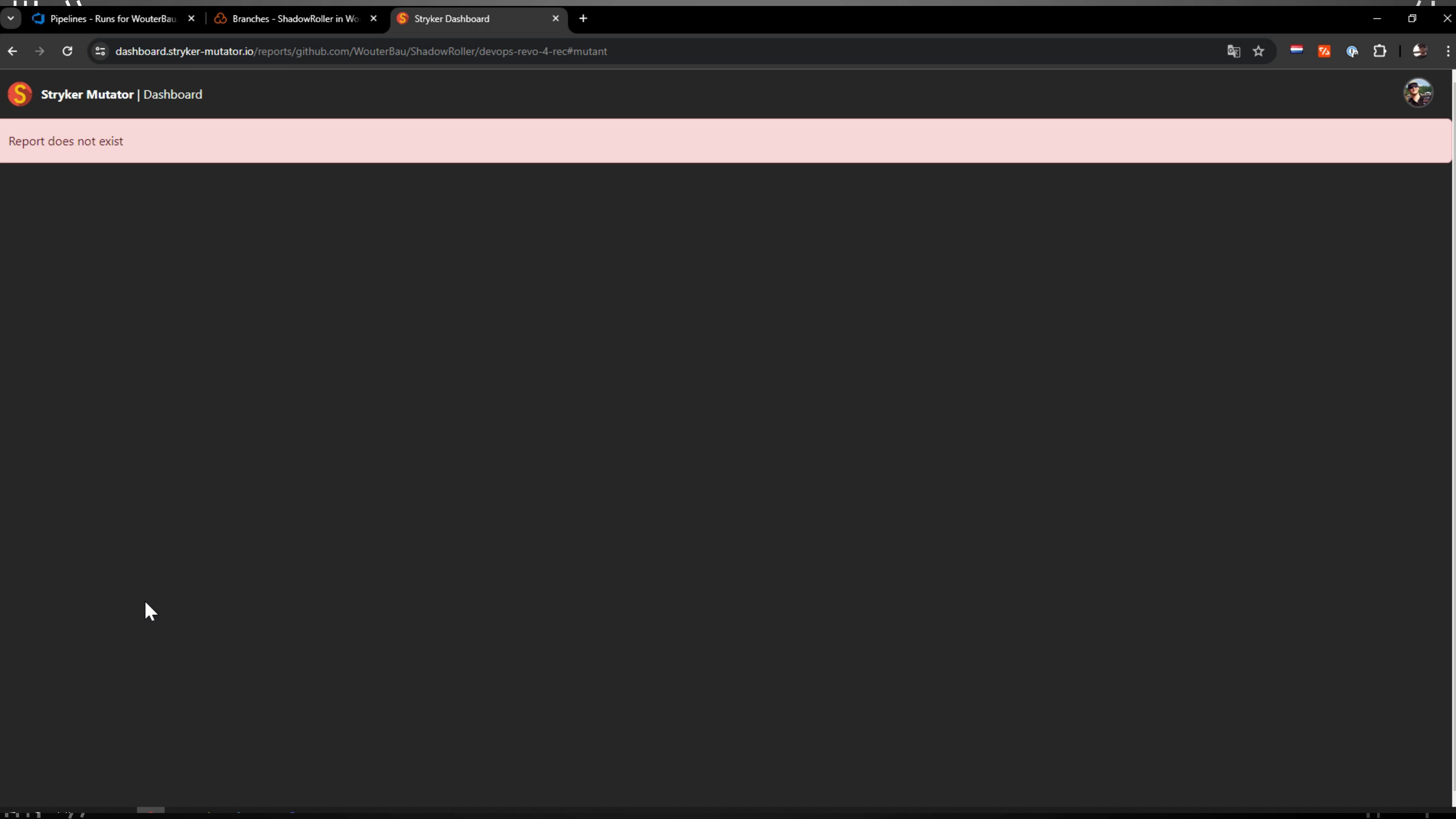

.azure-pipelines > qa-azure-pipelines.yml > [ ] jobs > {} 0 > [ ] steps
ryFileLocation

```yaml
jobs:
- job: sast
  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'      You, 1 hour ago • Merge branch 'temp/test-dockerized-azdevops'

- 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'
```

SHADOWROLLER



```dockerfile
#✨Run the tests
RUN dotnet test ShadowRoller.sln --no-build --logger trx --results-directory ./TestResults/ /
p:CollectCoverage=true /p:CoverletOutputFormat=json%2ccobertura%2copencover /
p:CoverletOutput=../../TestResults/ /p:MergeWith=../../TestResults/coverage.json /
p:SkipAutoProps=true /m:1      You, 1 hour ago • Merge branch 'temp/test-dockerized-azdevops'
```


Pipelines - Runs for WouterBau
Branches - ShadowRoller in Wo
Stryker Dashboard
dashboard.stryker-mutator.io/reports/github.com/WouterBau/ShadowRoller/devops-revo-4-rec#mutant
Stryker Mutator | Dashboard

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 here soon: https://bauchart.be

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