



Code Coverage Configuration

Docker File Changes

Overview

This documentation covers the transition to a **Bookworm-slim** build. To enable code coverage, the standard runtime image is modified into a "hybrid" image that contains both the .NET SDK and the production runtime. Additionally, a specialized **entrypoint script** is introduced to manage the process lifecycle, ensuring that the application runs under instrumentation and shuts down gracefully without losing coverage data.

1. Description of Dockerfile Additions

- **SDK & Runtime Convergence:** Instead of using separate **AS build** and **AS runtime** final stages, this file installs the **.NET SDK 8.0** directly onto the **8.0-bookworm-slim** base. This allows the use of **dotnet tool** to install the coverage agents.
- **Bookworm Native Stack:** It installs **libc6**, **libssl3**, and **libgssapi-krb5-2** specifically for Debian 12. It also includes **procps** (for the **ps** command) and **coreutils**, which are necessary for the background process management used in the entrypoint.
- **Tooling Path:** The environment variable **PATH="\$PATH:/root/.dotnet/tools"** is set so that the system can locate **dotnet-coverage** globally.
- **Script Injection:** The local **entrypoint.sh** is copied into the image and granted execution permissions (**chmod +x**).

2. Description of Entrypoint Script Logic

The **entrypoint.sh** script is the "brain" of the container at runtime.

- **PID 1 Management:** By running as the entrypoint, the script becomes Process ID 1. It is responsible for reaping "zombie" processes and handling system signals.

3. Implementation Files

In the sections below, the code added specifically for **Code Coverage changes to be added in runtime or final stage** is highlighted in **yellow**.

A. The Dockerfile

```
# RUN ALL CONTAINERS FROM ROOT (folder with .sln file):
# docker-compose build
# docker-compose up
#
# RUN JUST THIS CONTAINER FROM ROOT (folder with .sln file):
# docker build --pull -t web -f src/Web/Dockerfile .
#
# RUN COMMAND
# docker run --name eshopweb --rm -it -p 5106:5106 web
FROM mcr.microsoft.com/dotnet/sdk:8.0-bookworm-slim AS build
WORKDIR /app

COPY *.sln ./
COPY . .
WORKDIR /app/src/Web
RUN dotnet restore

RUN dotnet publish -c Release -o out

FROM mcr.microsoft.com/dotnet/aspnet:8.0-bookworm-slim AS runtime
WORKDIR /app
ENV COVERAGE_SESSION_ID=cavisson
ENV DEBIAN_FRONTEND=noninteractive

# -----
# OS dependencies
# -----
RUN apt-get update && apt-get install -y --no-install-recommends \
  ca-certificates \
  wget \
  curl \
  gnupg \
  libc6 \
  libc6-dev \
  libgcc-s1 \
  libstdc++6 \
```

Dockerfile and Entry point Changes for Code Coverage

```
libssl3 \  
zlib1g \  
libxml2 \  
libgssapi-krb5-2 \  
procps \  
coreutils \  
&& rm -rf /var/lib/apt/lists/*  
  
# -----  
# Install Microsoft repo  
# -----  
RUN wget https://packages.microsoft.com/config/debian/12/packages-microsoft-prod.deb -O  
packages-microsoft-prod.deb \  
&& dpkg -i packages-microsoft-prod.deb \  
&& rm packages-microsoft-prod.deb  
  
# -----  
# Install .NET SDK 8  
# -----  
RUN apt-get update && apt-get install -y --no-install-recommends \  
dotnet-sdk-8.0 \  
&& rm -rf /var/lib/apt/lists/*  
  
# -----  
# Install coverage tools  
# -----  
RUN dotnet tool install --tool-path /tools dotnet-coverage \  
  
&& dotnet tool install --tool-path /tools dotnet-reportgenerator-globaltool  
  
ENV PATH="/tools:${PATH}"  
  
COPY --from=build /app/src/Web/out ./  
  
COPY src/Web/entrypoint.sh /entrypoint.sh  
RUN chmod +x /entrypoint.sh
```

```
ENTRYPOINT ["/entrypoint.sh"]
```

B. The Entrypoint Script (`entrypoint.sh`)

```
Bash
```

```
#!/bin/sh
```

```
set -e
```

```
# Start app in background
```

```
dotnet {{application dll}}&
```

```
APP_PID=$!
```

```
# Signal Trap for Graceful Shutdown
```

```
_term() {
```

```
  echo "Entrypoint: TERM/INT received, forwarding to child..." >&2
```

```
  if [ -n "$APP_PID" ] && kill -0 "$APP_PID" 2>/dev/null; then
```

```
    kill -TERM "$APP_PID" 2>/dev/null || true
```

```
    wait "$APP_PID" 2>/dev/null || true
```

```
  fi
```

```
  exit 0
```

```
}
```

```
trap _term TERM INT
```

```
# Wait for App Exit
```

```
if [ -n "$APP_PID" ]; then
```

```
  wait "$APP_PID" 2>/dev/null
```

```
  echo "Entrypoint: child (pid $APP_PID) exited." >&2
```

```
  APP_PID=""
```

```
fi
```

```
# Keep container alive for data collection
```

```
tail -f /dev/null
```

Required Changes in `deployment.yaml` for supporting code coverage in `READ_ONLY` container

It will work with build number 4.14.1_31 onwards as there are some minor changes in the remote scripts to handle the "`COVERAGE_BASE_PATH`" variable.

Dockerfile and Entry point Changes for Code Coverage

volumes:

- name: tmp-writable
emptyDir: {}

volumeMounts:

- name: tmp-writable
mountPath: /tmp

containers:

- name: container_name
image:image_name:latest

securityContext:

readOnlyRootFilesystem: true

env:

- name: COVERAGE_BASE_PATH

value: "/tmp"