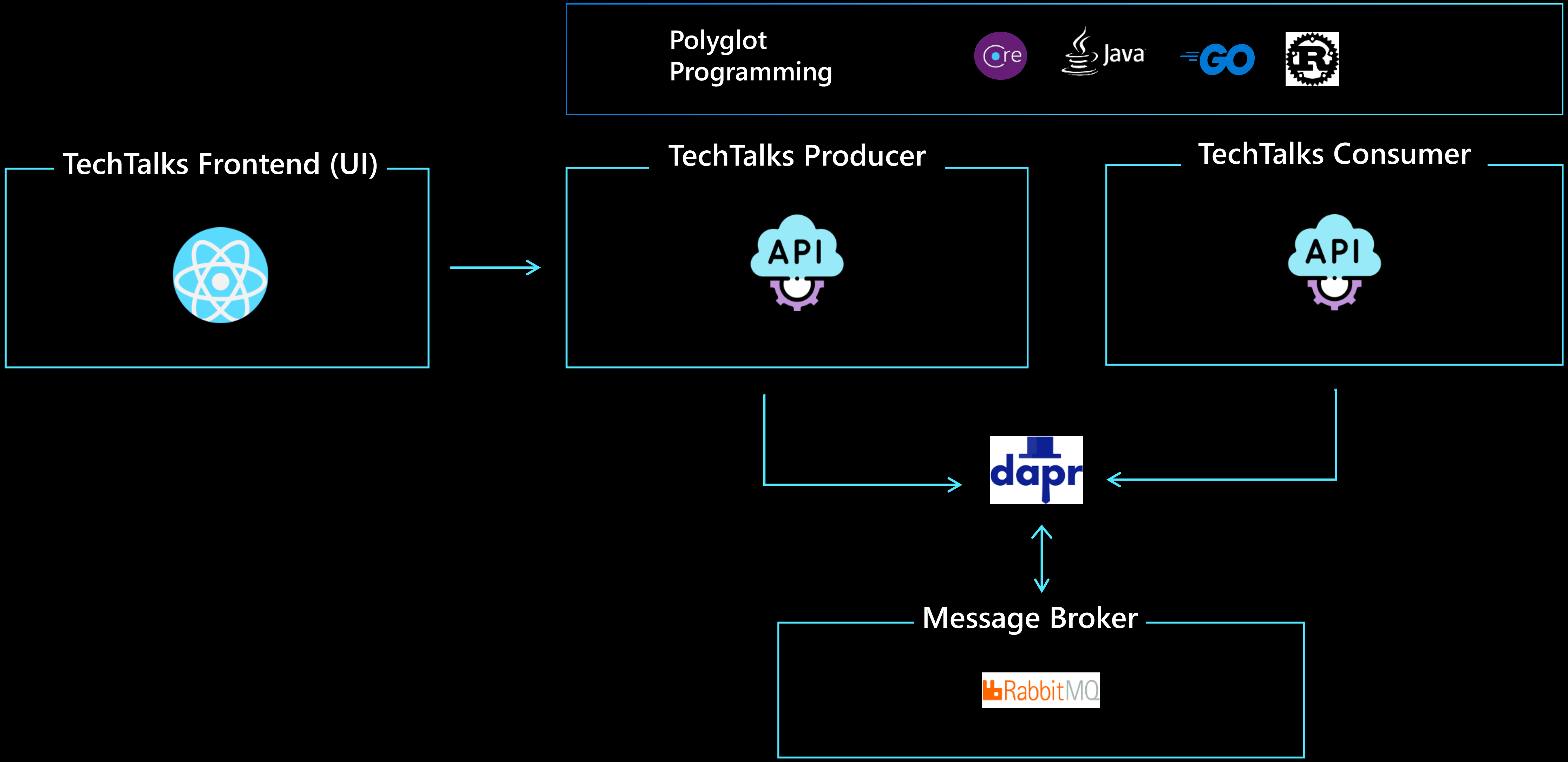
Cloud Native Ninja

Containerize DOTNET Apps

Nilesh Gule @nileshgule

TechTalks Application Architecture



Polyglot Programming

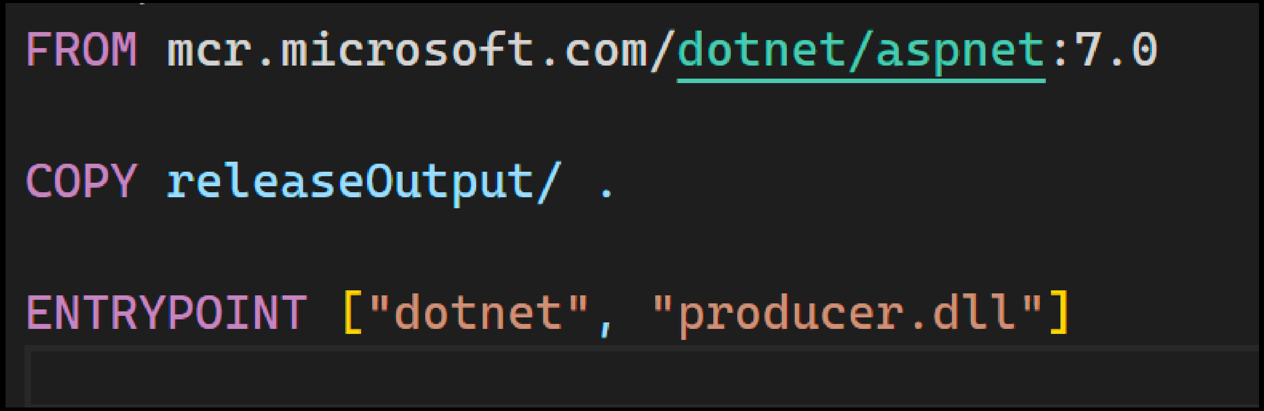
TechTalks Frontend (UI)

TechTalks Producer

TechTalks Consumer

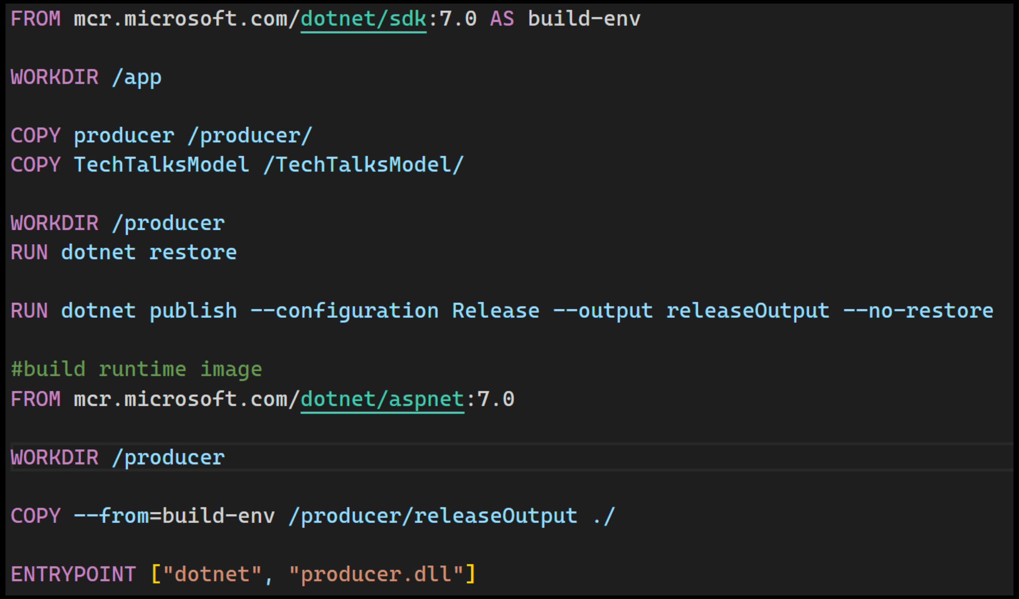
Message Broker

Dockerfile V1

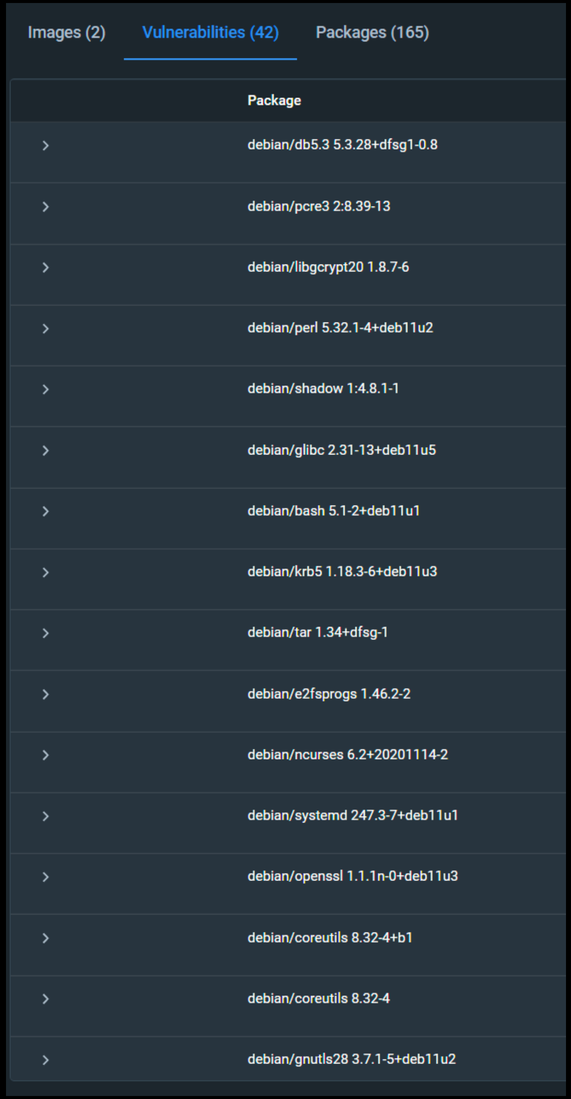


```
FROM mcr.microsoft.com/dotnet/aspnet:7.0

COPY releaseOutput/ .

ENTRYPOINT ["dotnet", "producer.dll"]
```

Dockerfile V2 – Multi stage builds

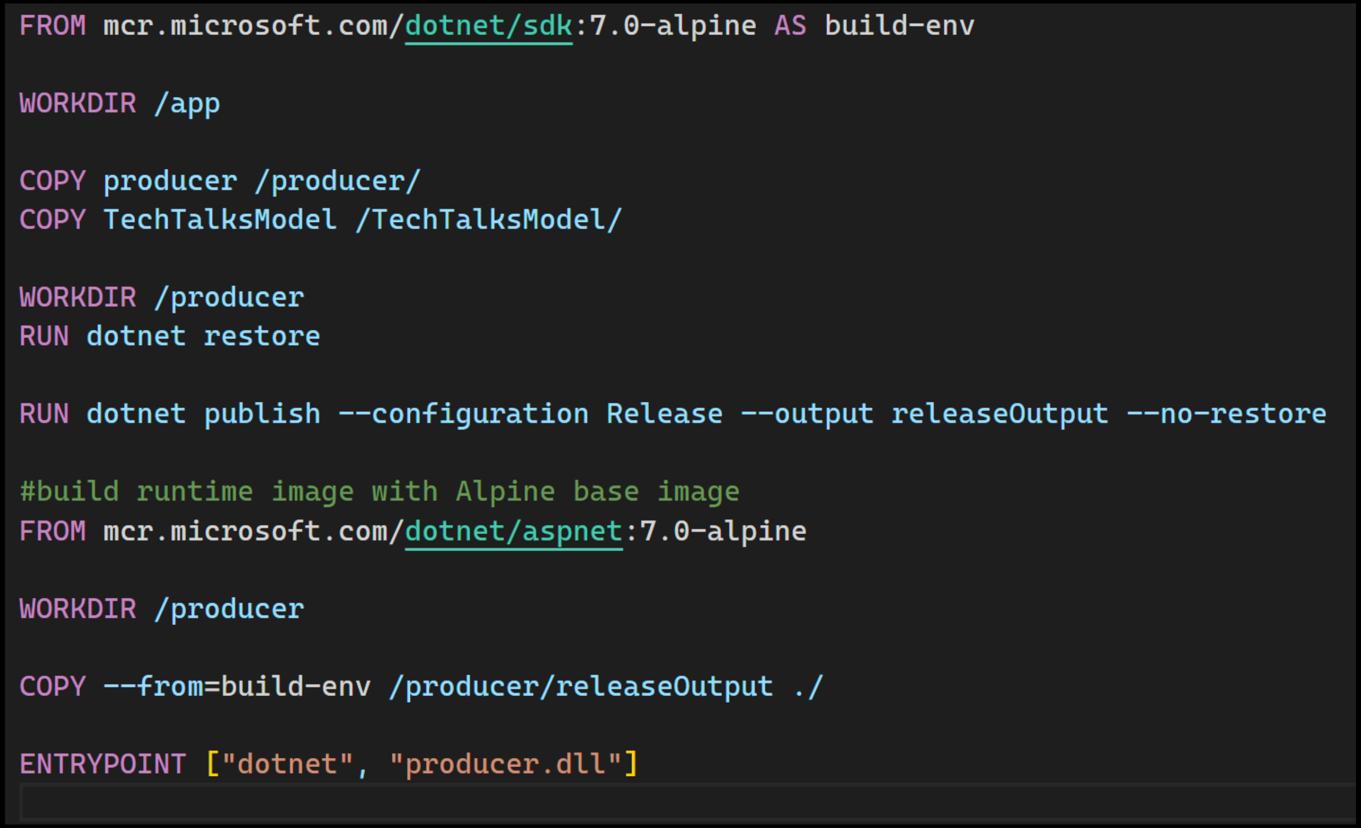


```
FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build-env

WORKDIR /app

COPY producer /producer/
COPY TechTalksModel /TechTalksModel/

WORKDIR /producer
RUN dotnet restore

RUN dotnet publish --configuration Release --output releaseOutput --no-restore

#build runtime image
FROM mcr.microsoft.com/dotnet/aspnet:7.0

WORKDIR /producer

COPY --from=build-env /producer/releaseOutput ./

ENTRYPOINT ["dotnet", "producer.dll"]
```

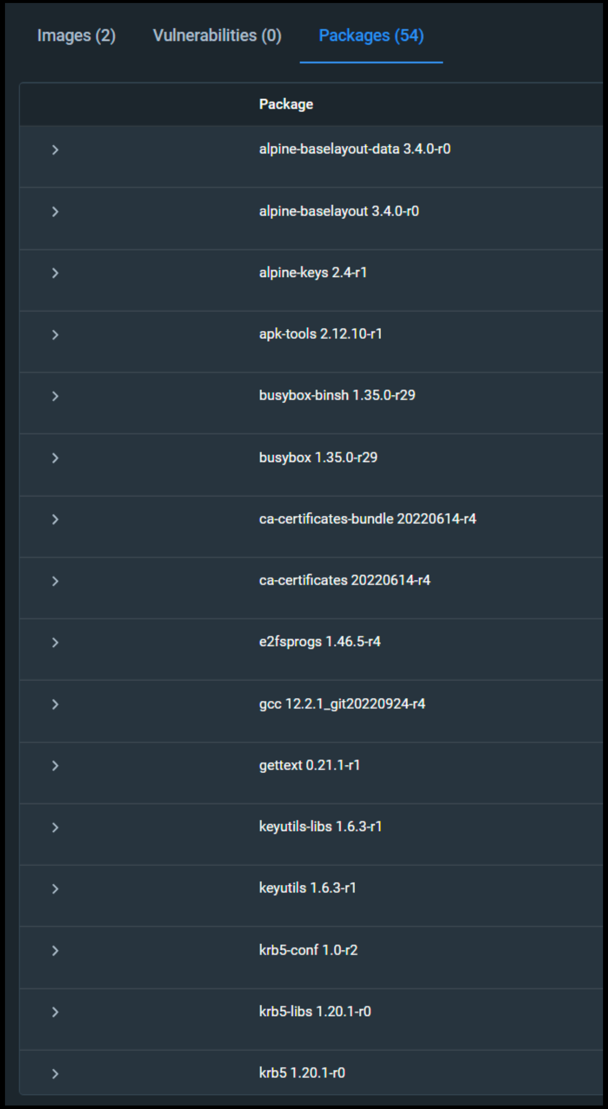


Package
debian/db5.3 5.3.28+dfsg1-0.8
debian/pcre3 2:8.39-13
debian/libgcrypt20 1.8.7-6
debian/perl 5.32.1-4+deb11u2
debian/shadow 1:4.8.1-1
debian/glibc 2.31-13+deb11u5
debian/bash 5.1-2+deb11u1
debian/krb5 1.18.3-6+deb11u3
debian/tar 1.34+dfsg-1
debian/e2fsprogs 1.46.2-2
debian/ncurses 6.2+20201114-2
debian/systemd 247.3-7+deb11u1
debian/openssl 1.1.1n-0+deb11u3
debian/coreutils 8.32-4+b1
debian/coreutils 8.32-4
debian/gnutls28 3.7.1-5+deb11u2

Dockerfile V3 – Multi stage builds Alpine image

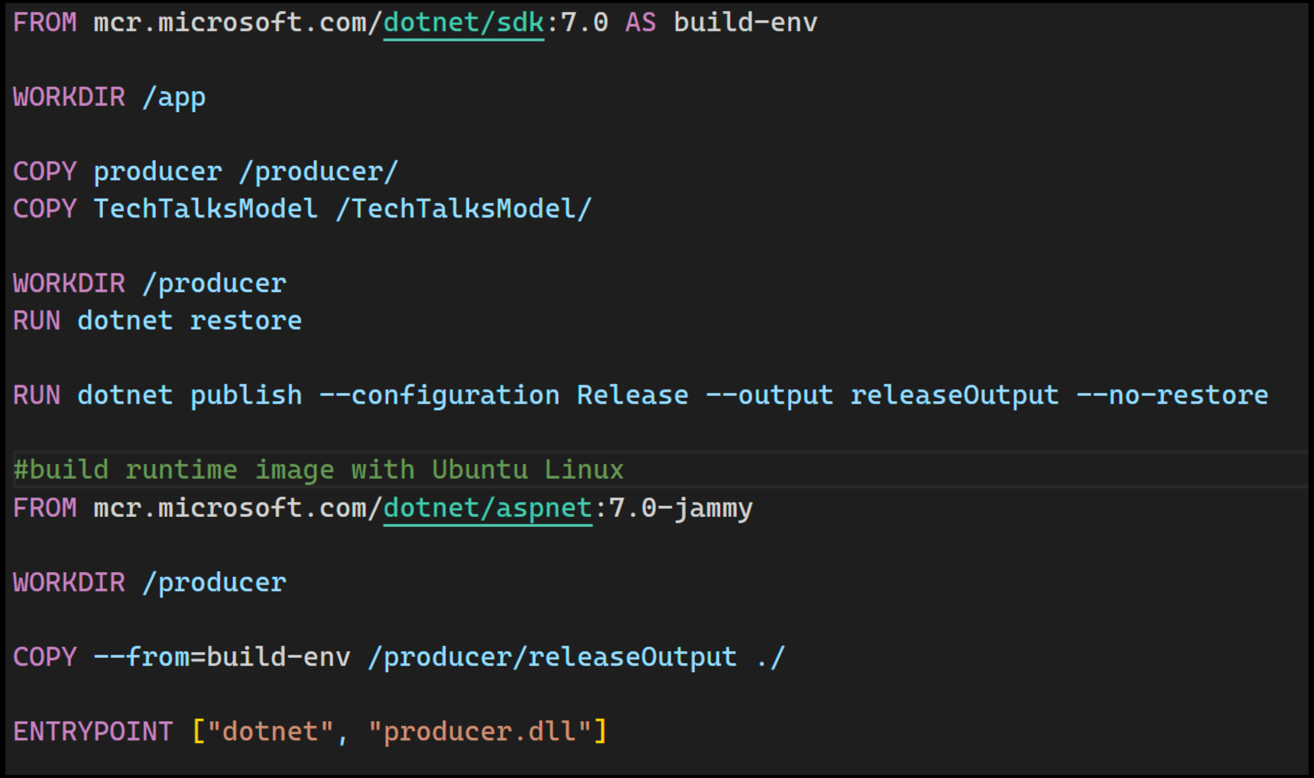


```
FROM mcr.microsoft.com/dotnet/sdk:7.0-alpine AS build-env

WORKDIR /app

COPY producer /producer/
COPY TechTalksModel /TechTalksModel/

WORKDIR /producer
RUN dotnet restore

RUN dotnet publish --configuration Release --output releaseOutput --no-restore

#build runtime image with Alpine base image
FROM mcr.microsoft.com/dotnet/aspnet:7.0-alpine

WORKDIR /producer

COPY --from=build-env /producer/releaseOutput ./

ENTRYPOINT ["dotnet", "producer.dll"]
```

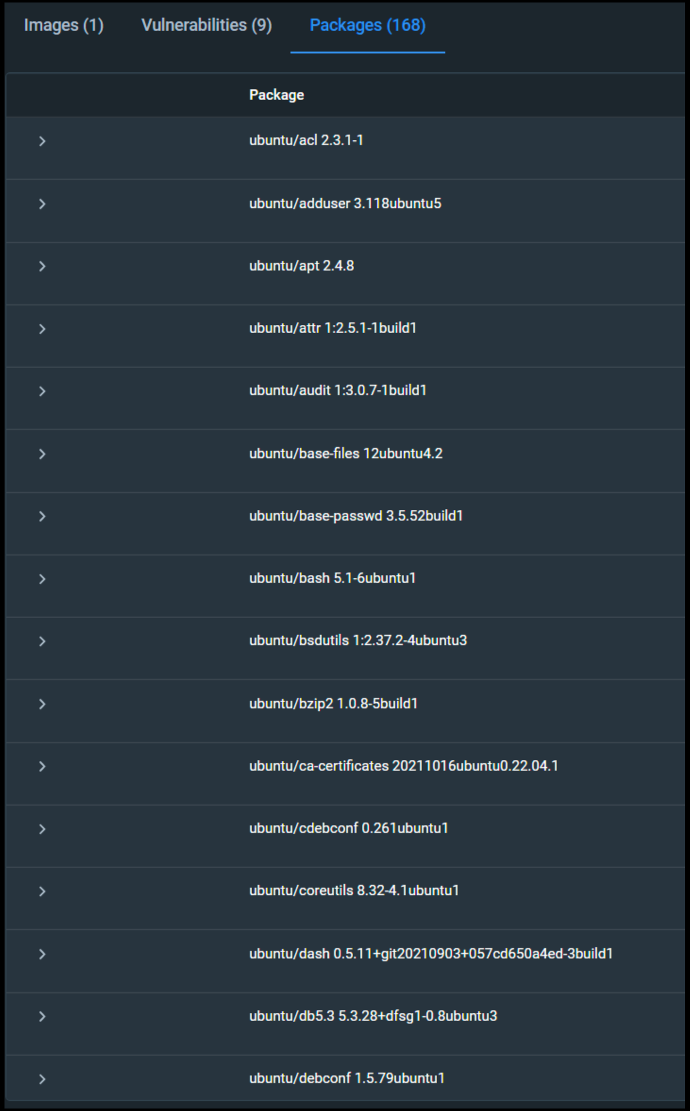


Package
alpine-baselayout-data 3.4.0-r0
alpine-baselayout 3.4.0-r0
alpine-keys 2.4-r1
apk-tools 2.12.10-r1
busybox-binsh 1.35.0-r29
busybox 1.35.0-r29
ca-certificates-bundle 20220614-r4
ca-certificates 20220614-r4
e2fsprogs 1.46.5-r4
gcc 12.2.1_git20220924-r4
gettext 0.21.1-r1
keyutils-libs 1.6.3-r1
keyutils 1.6.3-r1
krb5-conf 1.0-r2
krb5-libs 1.20.1-r0
krb5 1.20.1-r0

Dockerfile V4 – Multi stage builds Ubuntu image



```
FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build-env

WORKDIR /app

COPY producer /producer/
COPY TechTalksModel /TechTalksModel/

WORKDIR /producer
RUN dotnet restore

RUN dotnet publish --configuration Release --output releaseOutput --no-restore

#build runtime image with Ubuntu Linux
FROM mcr.microsoft.com/dotnet/aspnet:7.0-jammy

WORKDIR /producer

COPY --from=build-env /producer/releaseOutput ./

ENTRYPOINT ["dotnet", "producer.dll"]
```



	Package
›	ubuntu/acl 2.3.1-1
›	ubuntu/adduser 3.118ubuntu5
›	ubuntu/apt 2.4.8
›	ubuntu/attr 1:2.5.1-1build1
›	ubuntu/audit 1:3.0.7-1build1
›	ubuntu/base-files 12ubuntu4.2
›	ubuntu/base-passwd 3.5.52build1
›	ubuntu/bash 5.1-6ubuntu1
›	ubuntu/bsdutils 1:2.37.2-4ubuntu3
›	ubuntu/bzip2 1.0.8-5build1
›	ubuntu/ca-certificates 20211016ubuntu0.22.04.1
›	ubuntu/cdebconf 0.261ubuntu1
›	ubuntu/coreutils 8.32-4.1ubuntu1
›	ubuntu/dash 0.5.11+git20210903+057cd650a4ed-3build1
›	ubuntu/db5.3 5.3.28+dfsg1-0.8ubuntu3
›	ubuntu/debconf 1.5.79ubuntu1

Run Containerized App

Interactive terminal mode

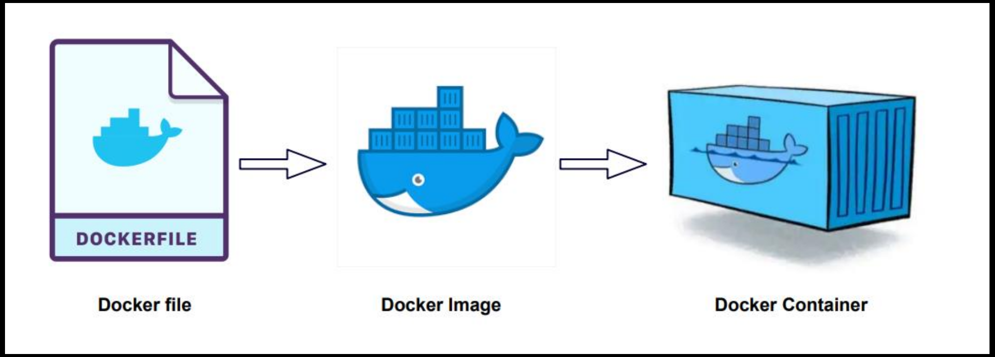
```
  docker run --rm `
  -it `
1 --name producer `
  -p 5000:80 ` 2
3 -e ASPNETCORE_ENVIRONMENT=Development `
  producer:v1  4
```

Detached mode

```
docker run --rm `
-d `
--name producer `
-p 5000:80 `
-e ASPNETCORE_ENVIRONMENT=Development `
producer:v1
```

- ✓ **1**: container name
- ✓ **2**: 5000 host port, 80 container port
- ✓ **3**: environment variable ASPNETCORE_ENVIRONMENT
- ✓ 4: image name and tag, default tag = latest

Summary



- ✓ **Base images**: Provided by major tools vendors, programming languages and frameworks
- ✓ **Dockerfile**: set of instructions to package the application with all its dependencies
- ✓ **Multi Stage build**: reduce image sizes with optimized base images for build and run stages
- ✓ Always use the best possible base image for the app to keep the size small and reduce attack surface

Containerize Apps Resources

Cloud Native Ninja GitHub repo: https://github.com/NileshGule/cloud-native-ninja

Slides

Slideshare: https://www.slideshare.net/nileshgule/

Speaker Deck: https://speakerdeck.com/nileshgule/