



# Building Multiplatform Docker Images for Rexx Distributions

René Vincent Jansen  
33rd Internationals Rexx Language  
Symposium, 14 September 2022







# Agenda

1 Why Containers

2 Building a Container

3 A multiplatform image

4 Docker Hub

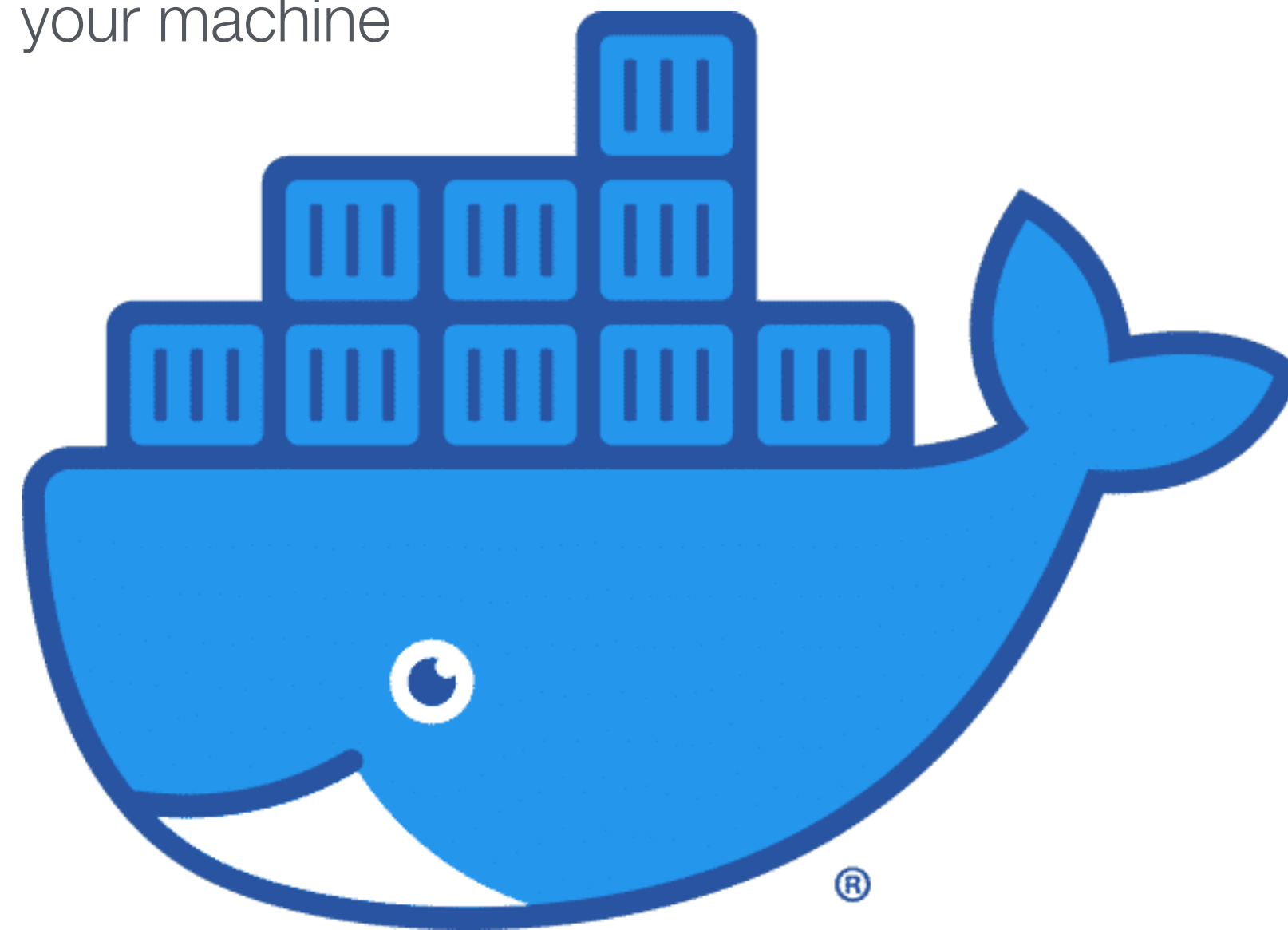


No 1

Why containers?

# What is a container

- Started many years ago as Unix chroot *jail*\* environments
- Is an important cloud distribution mechanism
- Will help you at home to save time to spend on other people
- Isolated environment on your machine



»» \* Isolating a process from all resources except where explicitly allowed



# Which problems does it solve?

- It virtualises - run different operating environments (OS, Instruction Sets) on the same platform
- It isolates - no more conflicting versions of libraries
- It packages - an application can be delivered installed and ready
  - Take for example a packaged VM/370CE - download and run it right away on your laptop or home system
- It protects: your running installations and software, excellent for testing and development
  - Want to test that new NetRexx version with your application? Just run the container
- Improving portability: develop for all (most) platforms



# How to use a container

- Install Docker Desktop (for macOS or Windows) or Docker (for Linux) (sometimes it is there already)

🔸 **docker run -it rvjansen/netrexx**

(First time only: it cannot find it and will download image)

```
→ ~ DOCKER RUN -IT RVJANSEN/NETREXX
ROOT@70205F7685D0:~# NRC
USAGE: /BIN/NETREXXC.SH [-RUN] [OTHER OPTIONS] FILENAME

NETREXX PORTABLE PROCESSOR 4.02-GA BUILD 55-20220124-1319
COPYRIGHT (C) REXXLA, 2011,2022.  ALL RIGHTS RESERVED.
PARTS COPYRIGHT (C) IBM CORPORATION, 1995,2008.

ARGUMENTS ARE: IN_FILE_SPECIFICATION... [-OPTION]...

USE "-HELP" TO SHOW ALL OPTIONS
ROOT@70205F7685D0:~# UNAME -A
LINUX 70205F7685D0 5.10.76-LINUXKIT #1 SMP PREEMPT Mon Nov 8 11:22:26 UTC 2021 AARCH64 GNU/LINUX
ROOT@70205F7685D0:~#
```

```
→ ~ DOCKER RUN RVJANSEN/NETREXX
UNABLE TO FIND IMAGE 'RVJANSEN/NETREXX:LATEST' LOCALLY
LATEST: PULLING FROM RVJANSEN/NETREXX
39AB78BC09E7: PULL COMPLETE
292AB8472872: PULL COMPLETE
E5670233BBC8: PULL COMPLETE
41D862CABD45: PULL COMPLETE
6760009C5245: PULL COMPLETE
DIGEST: SHA256:99235B08662351DD5B8638563733FFFD32AC83F95412B91E5BD38600348E48ED
STATUS: DOWNLOADED NEWER IMAGE FOR RVJANSEN/NETREXX:LATEST
→ ~
```



# Mount a local directory

- You have directories with program source in them
  - Your machine has NetRexx 4.0.4 and Java 19 installed but you see something funny and would like to retest it with NetRexx 4.02 and Java 8
  - So just run the corresponding Docker image, mount the source directory to that image and run the test. No install, no nothing.

```
docker run --rm -it -v "$PWD":/nrx -w /nrx rvjansen/netrexx
```

- This runs your container with the current directory mounted to /nrx in the image



No 2

Building a container

# You need a file called **Dockerfile**

Notice the long line to avoid several layers: each command produces a new virtual filesystem layer.

```
FROM debian
LABEL maintainer="rvjansen@xs4all.nl"
# Set the working directory
WORKDIR /

# Copy the current directory contents into the container at /
ADD nrws /bin

RUN apt-get update && \
    apt-get install apt-utils -y && \
    apt-get install readline-common -y && \
    apt-get install git -y && \
    apt-get install openjdk-11-jdk-headless -y && \
    apt-get install zip -y && \
    apt-get install wget -y && \
    wget http://netrexx.org/files/NetRexx-4.02-GA.zip && \
    unzip NetRexx-4.02-GA && \
    apt-get install nano -y && \
    apt-get install zsh -y && \
    mv /bin/pipe /bin/pipr && \
    chmod +x /bin/nrc && \
    chmod +x /bin/NetRexxC.sh && \
    chmod +x /bin/pipr && \
    echo 'alias pipc="java org.netrexx.njpipes.pipes.compiler"' >>/root/.bashrc && \
    echo 'alias pipe="java org.netrexx.njpipes.pipes.runner"' >>/root/.bashrc && \
    echo 'alias pipc="java org.netrexx.njpipes.pipes.compiler"' >>/root/.zshrc && \
    echo 'alias pipe="java org.netrexx.njpipes.pipes.runner"' >>/root/.zshrc

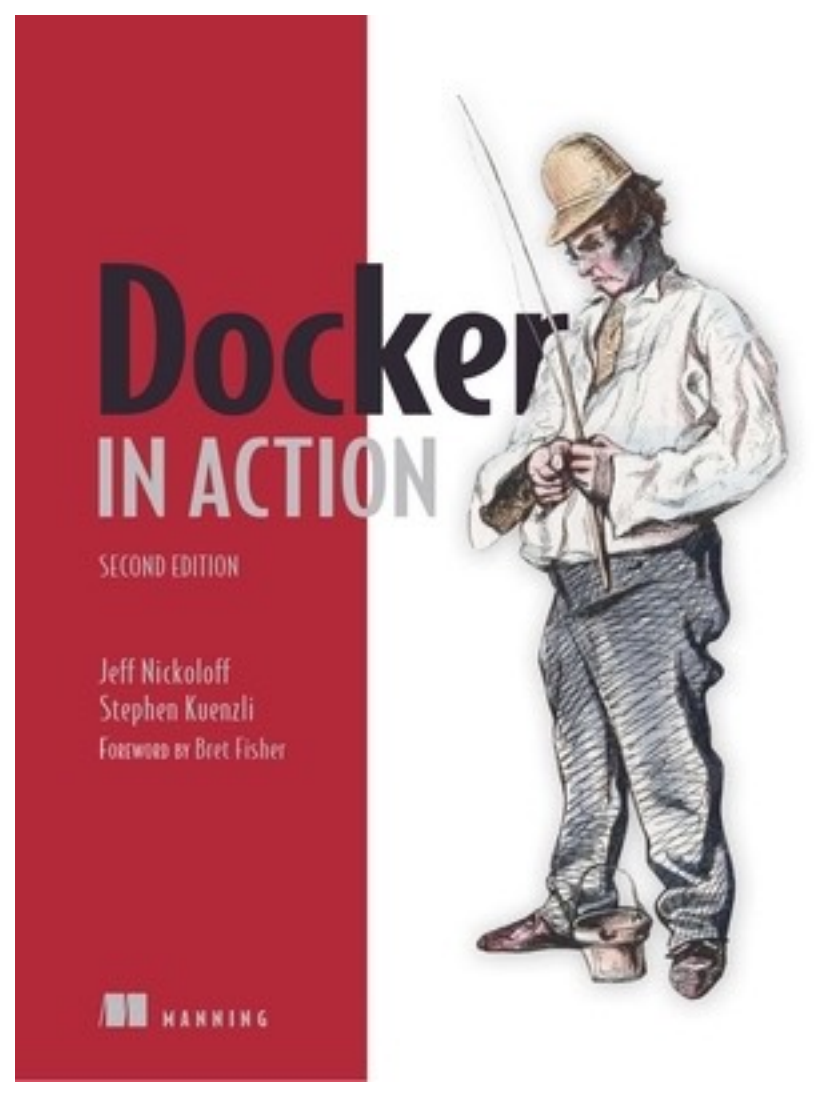
# Define environment variable
ENV CLASSPATH ./lib/NetRexxF.jar

# Run when the container launches
WORKDIR /root
```

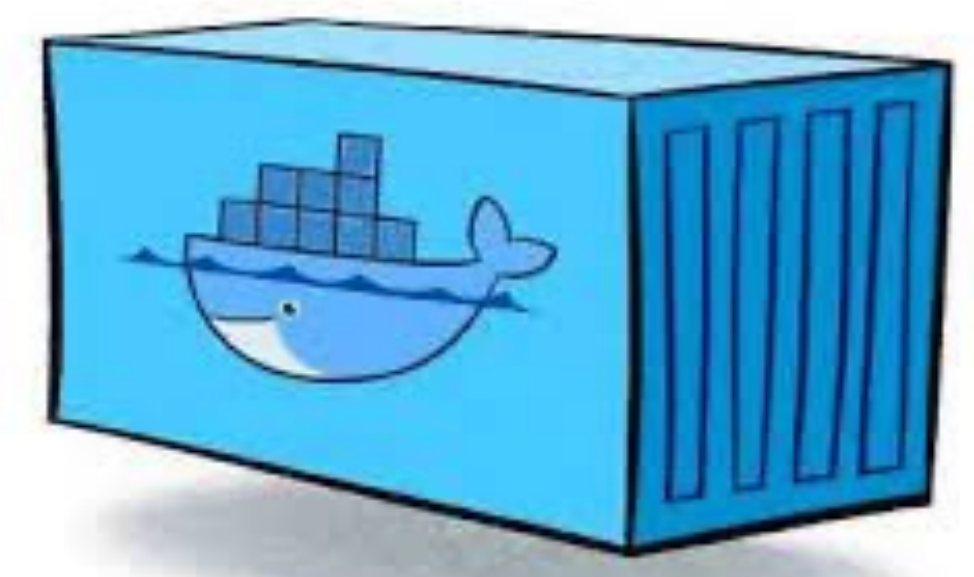


# Now build that image

```
❏ DOCKER BUILD --TAG RVJANSEN/NETREXX:LATEST .  
❏ DOCKER PUSH
```



(This is excellent documentation)



No 3

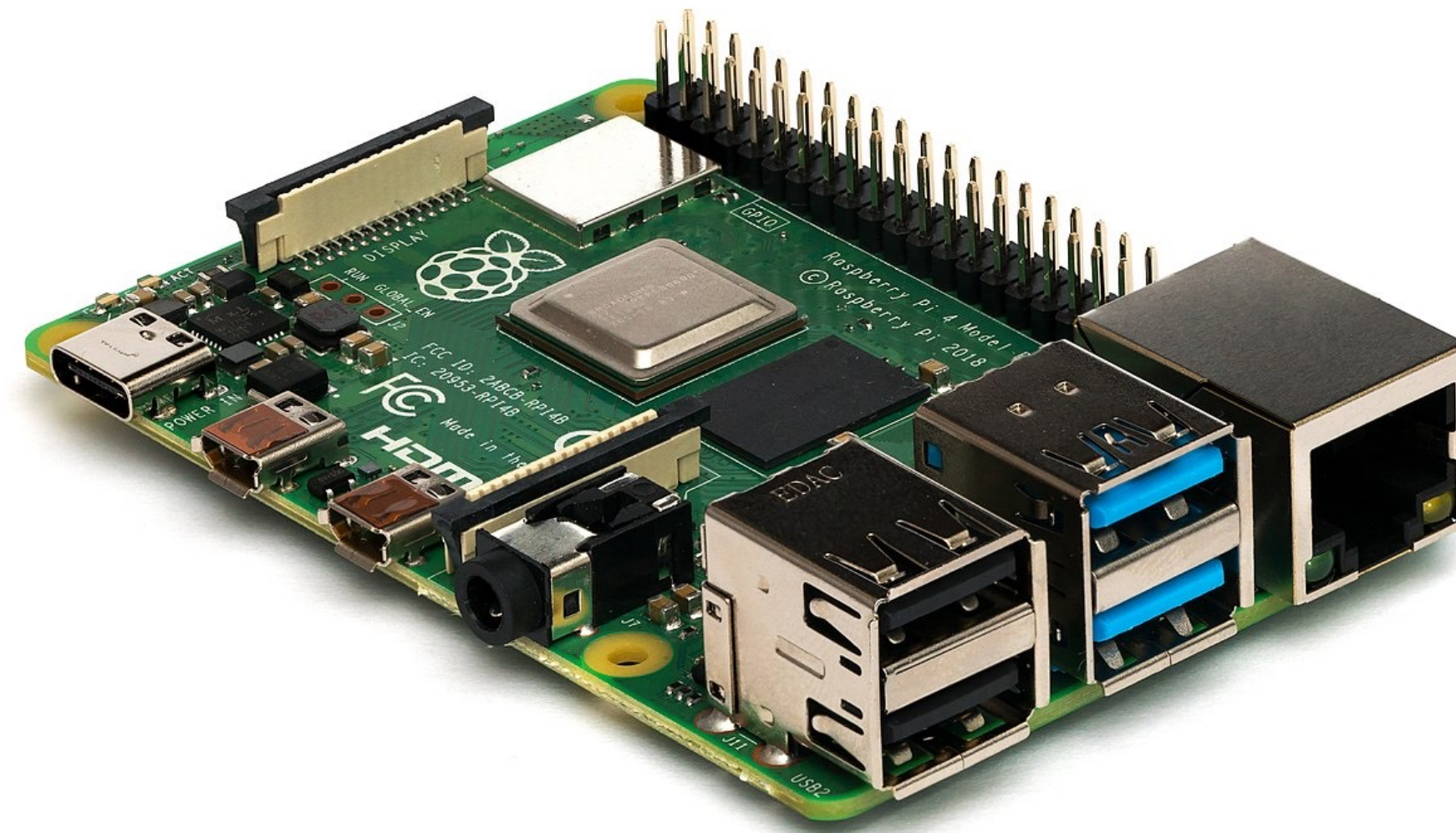
# A multiplatform Docker Image

Meaning, for multiple instruction set architectures



# Different machines have different machine instructions

(duh!)  
We call those ISA:  
Instruction Set  
Architecture





# Docker can run (or emulate) different ISA's

- The most efficient choice is to run the image on the native architecture
- Other architectures are emulated by **qemu**

- Architectures officially supported by Docker, Inc. for running Docker: (see [download.docker.com](https://download.docker.com))
  - ARMv6 32-bit ( `arm32v6` ): <https://hub.docker.com/u/arm32v6/>
  - ARMv7 32-bit ( `arm32v7` ): <https://hub.docker.com/u/arm32v7/>
  - ARMv8 64-bit ( `arm64v8` ): <https://hub.docker.com/u/arm64v8/>
  - Linux x86-64 ( `amd64` ): <https://hub.docker.com/u/amd64/>
  - Windows x86-64 ( `windows-amd64` ): <https://hub.docker.com/u/winamd64/>
- Other architectures built by official images: (but *not* officially supported by Docker, Inc.)
  - ARMv5 32-bit ( `arm32v5` ): <https://hub.docker.com/u/arm32v5/>
  - IBM POWER8 ( `ppc64le` ): <https://hub.docker.com/u/ppc64le/>
  - IBM z Systems ( `s390x` ): <https://hub.docker.com/u/s390x/>
  - MIPS64 LE ( `mips64le` ): <https://hub.docker.com/u/mips64le/>
  - RISC-V 64-bit ( `riscv64` ): <https://hub.docker.com/u/riscv64/>
  - x86/i686 ( `i386` ): <https://hub.docker.com/u/i386/>



# Choose a multiplatform base image

## Building an image with more architecture layers

First, a local copy of the buildx tool must be created, because the standard version can only build for the current architecture.

```
docker buildx create --use
```

This only needs to happen once per install of Docker.

Then the commandline for building the image for multiple architectures is:

```
docker buildx build --platform linux/amd64,linux/arm64,linux/arm/v7,linux/s390x -t  
rvjansen/netrexx:latest --push .
```

This will in fact build three images of which the layers are put together later, but these layers are only delivered for the requested architecture, which defaults to the instruction set the machine is running.

# Inspect the built image, see the arch layers

→ netrexx-code git:(master) ✕ docker buildx imagetools inspect rvjansen/netrexx:latest

Name: docker.io/rvjansen/netrexx:latest

MediaType: application/vnd.docker.distribution.manifest.list.v2+json

Digest: sha256:21ec4908ccced5496afa581c382309232262ef800600d42f9d5d58dae96efdf1

## Manifests:

Name: docker.io/rvjansen/netrexx:latest@sha256:71b41b19ce2ef12d3dd6c5f1d056730b1b1b8785168e51f896cddc8c23b24711

MediaType: application/vnd.docker.distribution.manifest.v2+json

Platform: linux/amd64

Name: docker.io/rvjansen/netrexx:latest@sha256:e618059472422c2c585b10ec2a0ca6149ae28b505d61310797fc2abfd037613c

MediaType: application/vnd.docker.distribution.manifest.v2+json

Platform: linux/arm64

Name: docker.io/rvjansen/netrexx:latest@sha256:38a9d767ff2861e0d1733906e1392ad94de587909f7aeb8afbe28d56ca22f929

MediaType: application/vnd.docker.distribution.manifest.v2+json

Platform: linux/arm/v7

Name: docker.io/rvjansen/netrexx:latest@sha256:03763fbd2aa74e715a08d87eb3a89c9552517c1aca85d0e42f8b717609537744

MediaType: application/vnd.docker.distribution.manifest.v2+json

Platform: linux/s390x





# Run the image in a container with another ISA

For example:

```
docker run -it rvjansen/  
netrexx:latest@sha256:03763fbd2aa74e715a08d87eb3a89c9552517c1aca8  
5d0e42f8b717609537744
```

This runs the s390x version, and starts by downloading the missing layer  
once

```
→ NETREXX-CODE GIT:(MASTER) X DOCKER RUN -IT RVJANSEN/NETREXX:LATEST@SHA256:03763FBD2AA74E715A08D87EB3A89C9552517C1ACA85D0E42F  
8B717609537744  
UNABLE TO FIND IMAGE 'RVJANSEN/NETREXX:LATEST@SHA256:03763FBD2AA74E715A08D87EB3A89C9552517C1ACA85D0E42F8B717609537744' LOCALLY  
DOCKER.IO/RVJANSEN/NETREXX@SHA256:03763FBD2AA74E715A08D87EB3A89C9552517C1ACA85D0E42F8B717609537744: PULLING FROM RVJANSEN/NETRE  
XX  
CF58BFABF9FB: PULL COMPLETE  
A2A78F34DFF8: PULL COMPLETE  
E37EEDC862D7: PULL COMPLETE  
2EBE9A1C5D82: PULL COMPLETE  
CF263D242024: PULL COMPLETE  
4F4FB700EF54: PULL COMPLETE  
DIGEST: SHA256:03763FBD2AA74E715A08D87EB3A89C9552517C1ACA85D0E42F8B717609537744  
STATUS: DOWNLOADED NEWER IMAGE FOR RVJANSEN/NETREXX@SHA256:03763FBD2AA74E715A08D87EB3A89C9552517C1ACA85D0E42F8B717609537744  
WARNING: THE REQUESTED IMAGE'S PLATFORM (LINUX/S390X) DOES NOT MATCH THE DETECTED HOST PLATFORM (LINUX/ARM64/V8) AND NO SPECIFIC  
C PLATFORM WAS REQUESTED  
ROOT@45c40ec82702:~# UNAME -A  
LINUX 45c40ec82702 5.10.76-LINUXKIT #1 SMP PREEMPT Mon Nov 8 11:22:26 UTC 2021 s390x GNU/LINUX  
ROOT@45c40ec82702:~#
```





No 4

# Docker Hub

<https://hub.docker.com>



rvjansen Search by repository name Create repository

- rvjansen / **netrexx** Not Scanned 0 stars 200 downloads Public Last pushed: 7 hours ago
- rvjansen / **4.02** Not Scanned 0 stars 16 downloads Public Last pushed: 8 months ago
- rvjansen / **lspf** Not Scanned 1 star 127 downloads Public Last pushed: 3 years ago
- rvjansen / **netrexx-build** Not Scanned 0 stars 29 downloads Public Last pushed: 3 years ago
- rvjansen / **arch-devel** Not Scanned 0 stars 42 downloads Public Last pushed: 4 years ago

Tip: Not finding your repository? Try switching namespace via the top left dropdown.

Organizations Create

- rexxla
- [View all...](#)



Community All-Hands: On-Demand

All sessions from our 6th Community All-Hands are now available on-demand! Over 35 talks cover best practices, demos, open source, product updates, community news, and more. Catch up on the sessions you missed — or review your favorites.

[Watch Now](#)





hub.docker.com/repository/docker/rvjansen/netrexx

Docker Hub

Failed to open page

docker hub Search for great content (e.g., mysql) Explore Repositories Organizations Help rvjansen

rvjansen Repositories netrexx

General Tags Builds Collaborators Webhooks Settings

### rvjansen / netrexx

**Description**  
 NetRexx on Debian + JDK 11 image includes pipes processor includes Workspace for NetRexx

Last pushed: 7 hours ago

**Docker commands** [Public View](#)

To push a new tag to this repository,

```
docker push rvjansen/netrexx:tagname
```

**Tags and scans** VULNERABILITY SCANNING - DISABLED [Enable](#)

This repository contains 7 tag(s).

TAG	OS	LOG4SHELL	PULLED	PUSHED
latest			4 hours ago	7 hours ago
3.09			7 months ago	7 months ago
3.11			7 months ago	7 months ago
4.01			7 months ago	7 months ago
4.02			7 months ago	8 months ago

[See all](#) [Go to Advanced Image Management](#)

**Recent builds**  
 Link a source provider and run a build to see build results here.





## README

### 1. Work with a shell within the image

As producing data within the image generally is not recommended this also involves a bind-mounted directory, but you will work inside of the shell in the docker container and you can use all the tools provided in the image.

A suitable command line would be: `docker run --rm -it -v "$PWD":/nrx -w /nrx rvjansen/netrexx`

If you want to keep changes in the container (for example, when you added tools or configuration that are useful and need to go into a new image, based on this image), do not use the `--rm` switch. The docker documentation explains how to commit this container and tag its new image. The `rvjansen/netrexx:3.07` will be downloaded once from the docker hub, when it is not on the local machine yet. It will know it has been downloaded the next time you start this image. Also, it will detect when the image has been updated.

The `-it` option is needed for an interactive terminal session. the `-v` switch bind mounts the current directory into a directory `/nrx` in the image.

The available tools within the image are being worked on, and documentation will be produced later.

### 2. Compile or exec from a shell on your host machine

The term 'host machine' is used here to indicate the fact that the docker image runs a guest OS.

A suitable command line would be: (assuming you want to compile a class called `RSAnrx` in the local directory)

`docker run --rm -v "$PWD":/nrx -w /nrx rvjansen/netrexx nrc RSAnrx`

Here, `--rm` will make sure the container is not kept, the `-v` tells docker to bind mount the current directory to a directory `/nrx` within the container, and `-w` sets this as the working directory. The `rvjansen/netrexx` will be downloaded once from the docker hub, when it is not on the local machine yet. It will know it has been downloaded the next time you start this image. Of course, in most shells is it possible to alias this command, or start a batchfile, c.q. a shell script containing this.



#### Explore

Containers

Pricing

#### Account

Content Subscriptions

Billing

#### Publish

Publisher Center

#### Resources

Docker Blog

Download Docker

#### Support

Feedback

Documentation

Hub Release Notes

Forums







Thank you!  
Questions?

