Dyalog's [Public] Docker Containers

Morten Kromberg

CTO, Dyalog Ltd.

Agenda

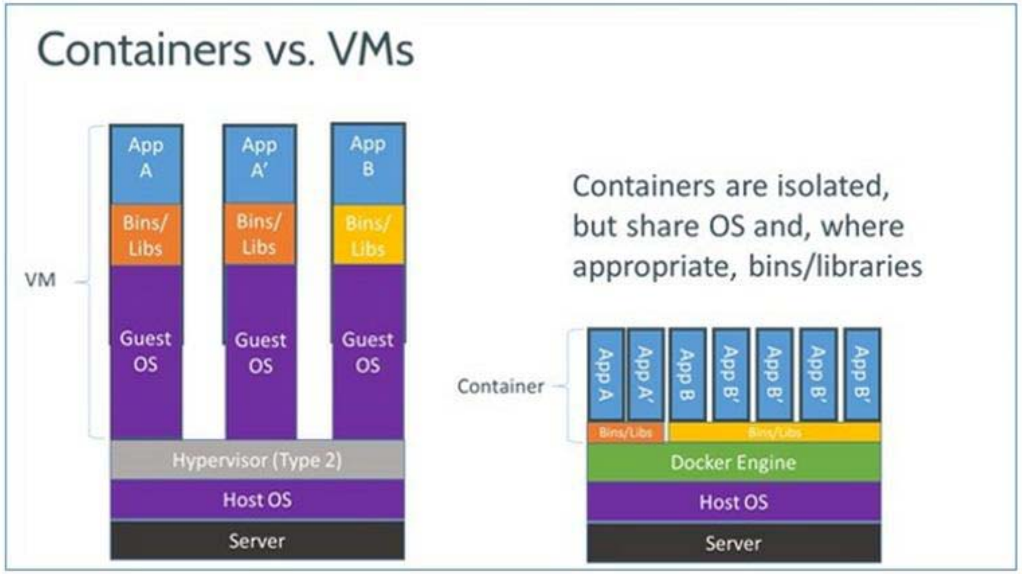
- What is a Container
  - Why should you want one?
- Dyalog's Public Docker Containers
  - Starting, Using, Stopping, Debugging
- Extending Dyalog's Containers
- Combining Containers using docker-compose

What is a "Container"?

Applications which run in a "Container" believe they are alone on a machine even though they are sharing it.

Similar to a "Virtual Machines" but does NOT require multiple copies of the O/S.

Containers & Docker

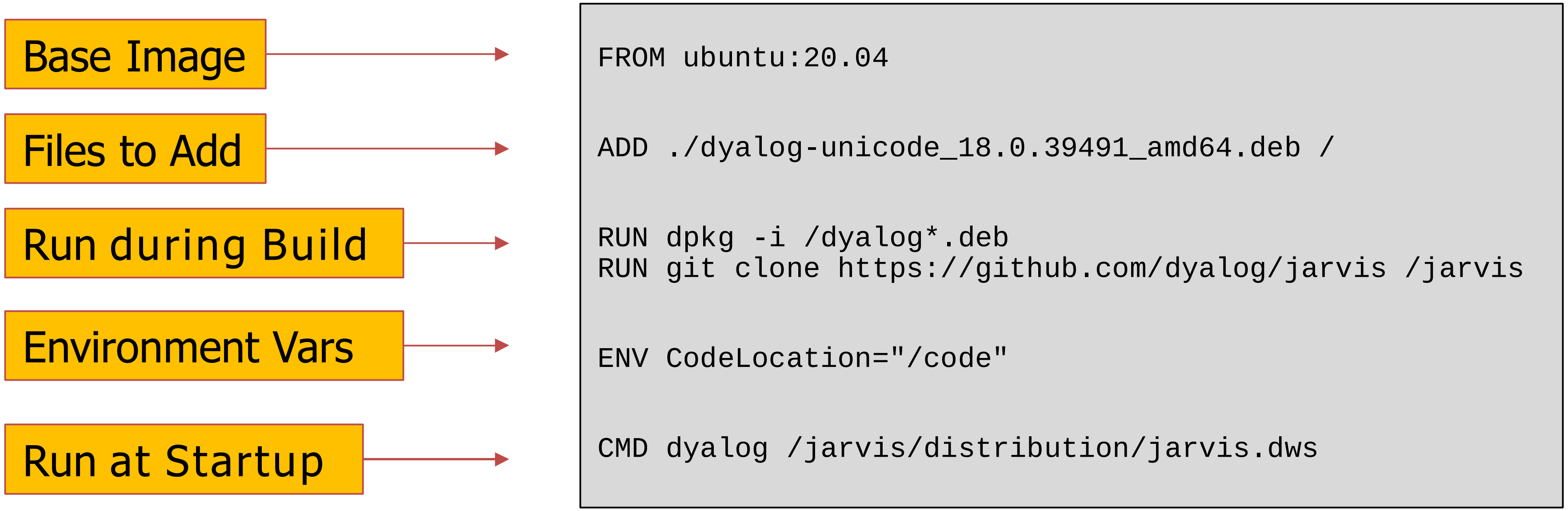


Containers vs. VMs

Containers are isolated, but share OS and, where appropriate, bins/libraries

http://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/

"Dockerfile" (text) describes the Container



Base Image

Files to Add

Run during Build

Environment Vars

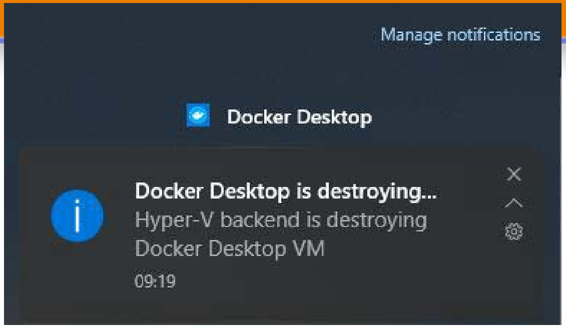
Run at Startup

```
FROM ubuntu:20.04

ADD ./dyalog-unicode_18.0.39491_amd64.deb /

RUN dpkg -i /dyalog*.deb
RUN git clone https://github.com/dyalog/jarvis /jarvis

ENV CodeLocation="/code"

CMD dyalog /jarvis/distribution/jarvis.dws
```

(Imaginary dyalog/jarvis image)

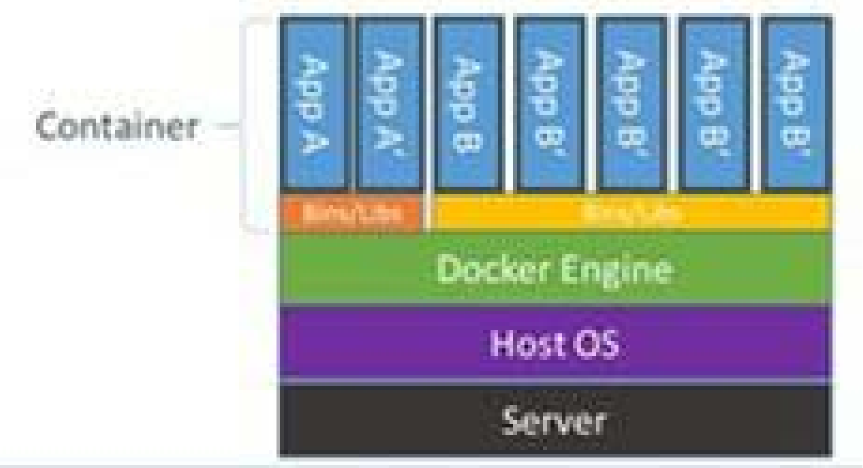
Linux or Windows?

- Most containers run Linux
  - Linux is smaller – more CPU and memory for your application: Less $£€
- The "Docker Engine" is available on Linux, macOS and Windows
- Recently, "Docker for Windows" switch to using WSL – Windows Subsystem for Linux


Manage notifications

Docker Desktop

Docker Desktop is destroying...
Hyper-V backend is destroying Docker Desktop VM
09:19



docker run … Starts the Container

```
docker run -it -p 8080:8080 dyalog/jarvis
```

`dyalog/jarvis` is the name of the image to start

docker run … allocate pseudo-TTY

```
docker run -it -p 8080:8080 dyalog/jarvis
```

(don't ask …
but also … don't forget!)

Switch	Description
`-p hhhh:cccc`	Map TCP port cccc in container to hhhh on host
`-e name=value`	Set environment variable inside the container
`-v /hfolder:/cfolder`	Mount hosts /hfolder as containers /cfolder
`-t`	Allocate a pseudo-TTY
`-i`	Keep stdin open even if not attached
`--rm`	Discard changes when container terminates

docker run cheat sheet

docker run … map TCP ports

```
docker run -it -p 80:8080 dyalog/jarvis
```

Maps port 8080 in container to 80 in the host

By default, the container network is completely isolated

Switch	Description
`-p hhhh:cccc`	Map TCP port cccc in container to hhhh on host
`-e name=value`	Set environment variable inside the container
`-v /hfolder:/cfolder`	Mount hosts /hfolder as containers /cfolder
`-t`	Allocate a pseudo-TTY
`-i`	Keep stdin open even if not attached
`--rm`	Discard changes when container terminates

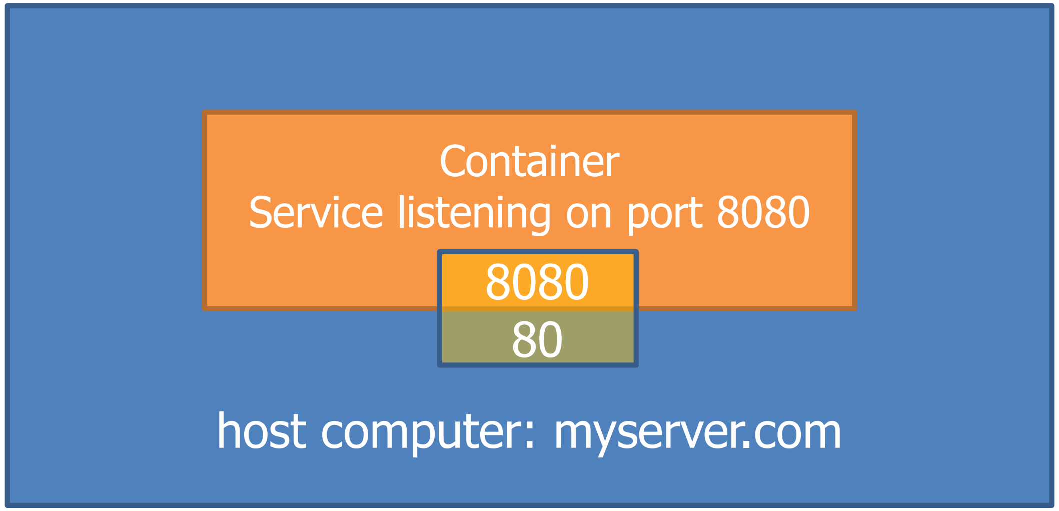
docker run cheat sheet

docker run … map TCP ports

```
docker run -it -p 80:8080 dyalog/jarvis
```

http://www.myserver.com

(web browser opens port **80**, which is mapped to **8080** in the container)



Container
Service listening on port 8080

host computer: myserver.com

docker run … mount volumes

```
docker run -it -v /my/app/folder:/code –p 80:8080 dyalog/jarvis
```

Mount the host folder /my/app/folder as /code inside the container.

Switch	Description
`-p hhhh:cccc`	Map TCP port cccc in container to hhhh on host
`-e name=value`	Set environment variable inside the container
`-v /hfolder:/cfolder`	Mount hosts /hfolder as containers /cfolder
`-t`	Allocate a pseudo-TTY
`-i`	Keep stdin open even if not attached
`--rm`	Discard changes when container terminates

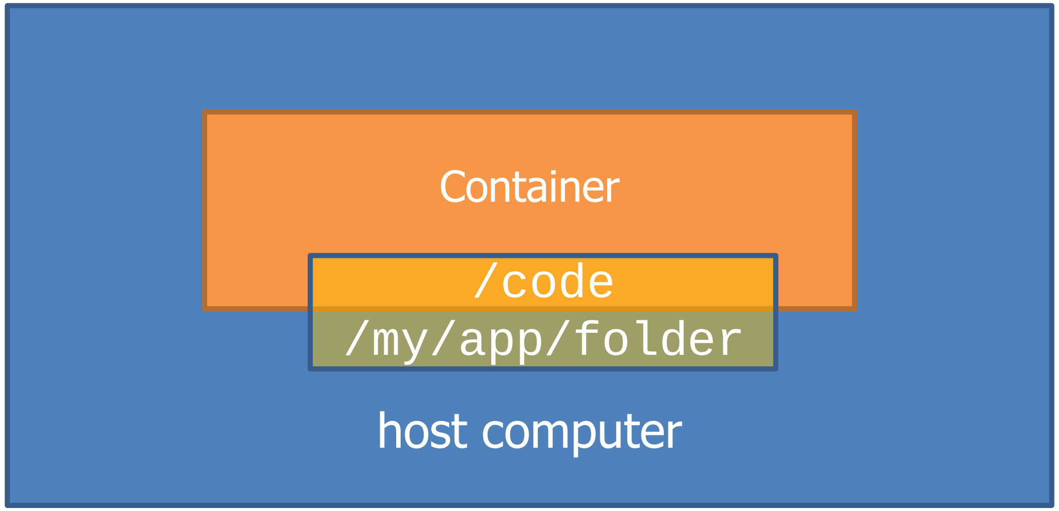
docker run cheat sheet

docker run … mount volumes

```
docker run -it -v /my/app/folder:/code –p 80:8080 dyalog/jarvis
```

The host folder /my/app/folder *is* /code inside the container.

(real-time mapping, updates made to /code in container persist after container stopped)



docker run … set environment vars

```
docker run -it -v /my/app/folder:/code -p 80:8080
-e RIDE_INIT=SERVE:*:4502 –p 4502:4502 dyalog/jarvis
```

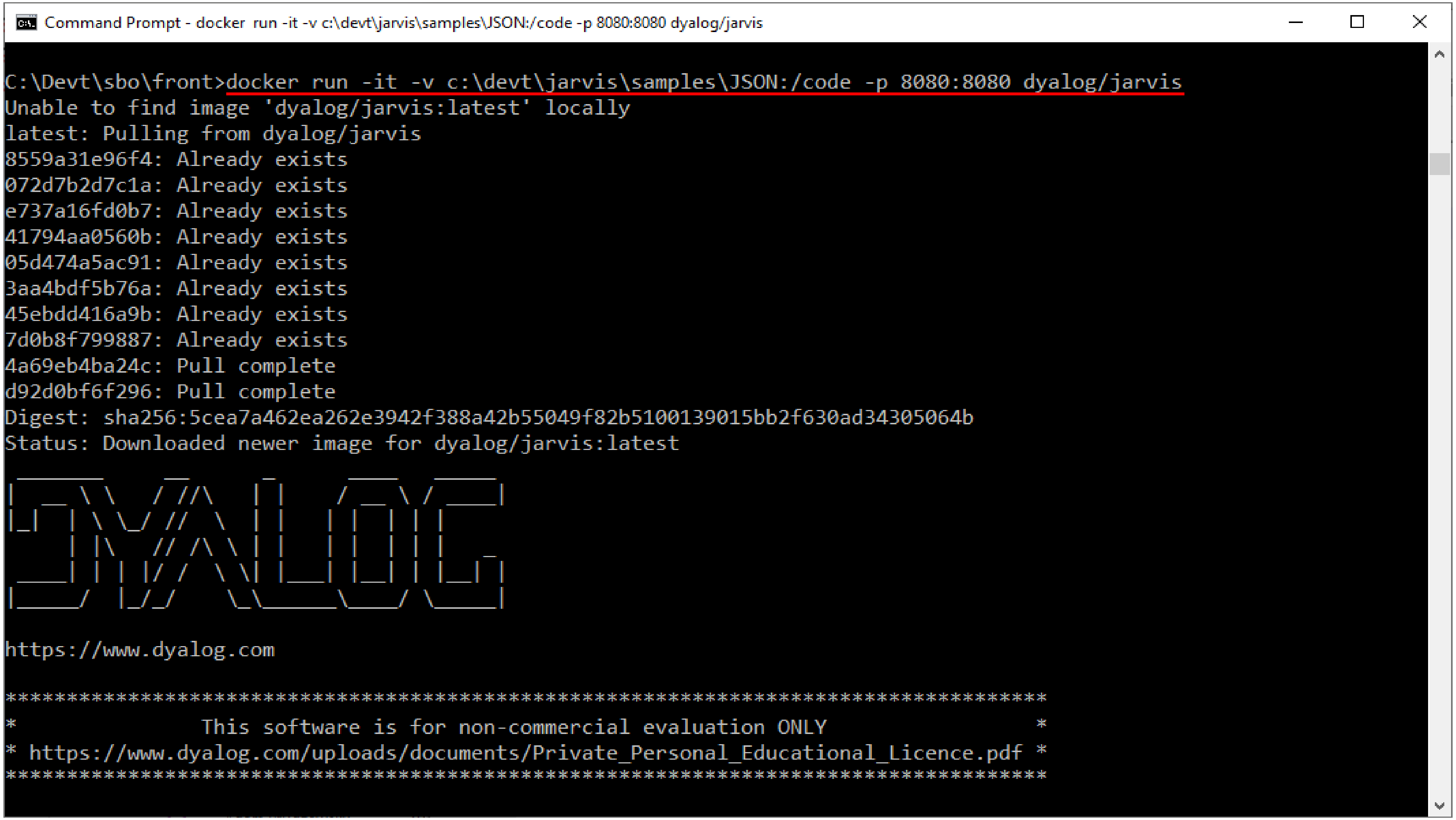
Set environment variable
RIDE_INIT
inside the container

NB: To connect to RIDE on port 4502 you MUST also
–p 4502:4502
or you will not be able to connect

Switch	Description
`-p hhhh:cccc`	Map TCP port cccc in container to hhhh on host
`-e name=value`	Set environment variable inside the container
`-v /hfolder:/cfolder`	Mount hosts /hfolder as containers /cfolder
`-t`	Allocate a pseudo-TTY
`-i`	Keep stdin open even if not attached
`--rm`	Discard changes when container terminates

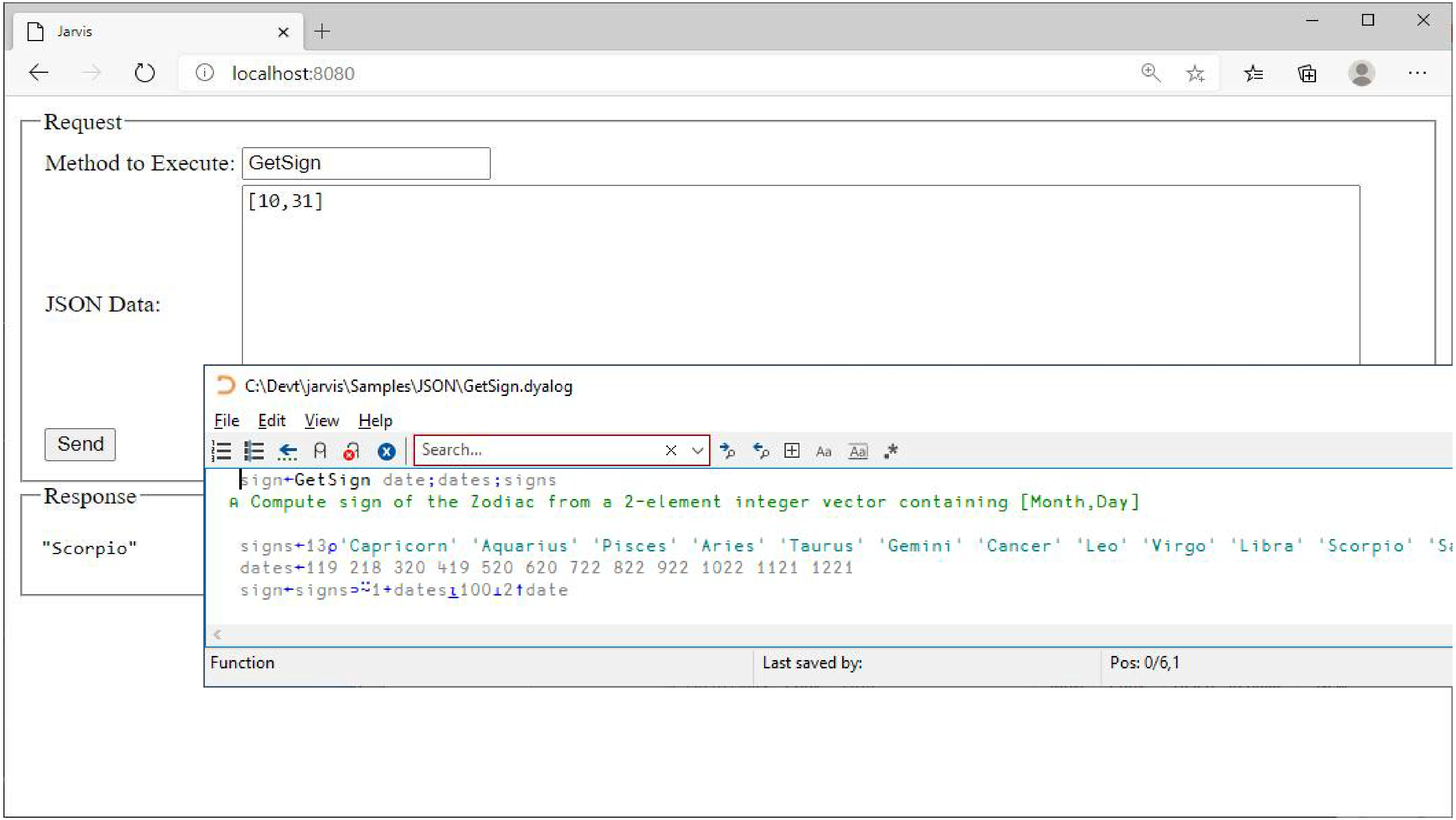
docker run cheat sheet

Demo 1 – Start & Use dyalog/jarvis



Command Prompt - docker run -it -v c:\devt\jarvis\samples\JSON:/code -p 8080:8080 dyalog/jarvis

```
C:\Devt\sbo\front>docker run -it -v c:\devt\jarvis\samples\JSON:/code -p 8080:8080 dyalog/jarvis
Unable to find image 'dyalog/jarvis:latest' locally
latest: Pulling from dyalog/jarvis
8559a31e96f4: Already exists
072d7b2d7c1a: Already exists
e737a16fd0b7: Already exists
41794aa0560b: Already exists
05d474a5ac91: Already exists
3aa4bdf5b76a: Already exists
45ebdd416a9b: Already exists
7d0b8f799887: Already exists
4a69eb4ba24c: Pull complete
d92d0bf6f296: Pull complete
Digest: sha256:5cea7a462ea262e3942f388a42b55049f82b5100139015bb2f630ad34305064b
Status: Downloaded newer image for dyalog/jarvis:latest
 _______     __     _       ____   _____
|  __ \ \   / //\   | |    / __ \ / ____|
|_|  | \ \_/ //  \  | |   | |  | | |
     | |\   // /\ \ | |   | |  | | |  _
 ____| | | |/ /   \ \| |___| |__| | |__| |
|_____/  |_/_/    \_\______\____/ \_____|

https://www.dyalog.com

*********************************************************************************
*               This software is for non-commercial evaluation ONLY             *
* https://www.dyalog.com/uploads/documents/Private_Personal_Educational_Licence.pdf *
*********************************************************************************
```



Request

Method to Execute: GetSign

JSON Data:

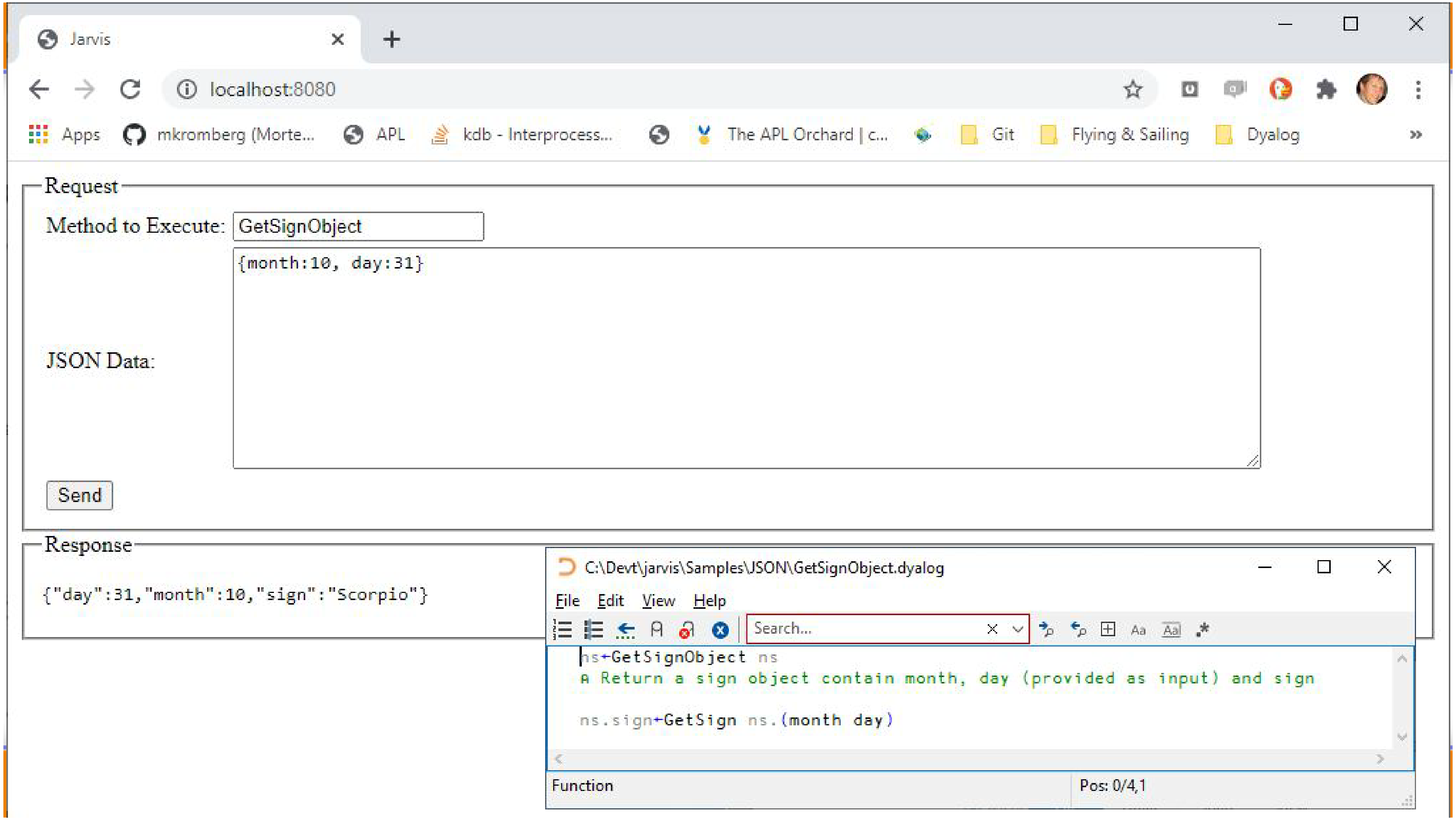
```
[10,31]
```

Send

Response

```
"Scorpio"
```

```
 sign←GetSign date;dates;signs
⍝ Compute sign of the Zodiac from a 2-element integer vector containing [Month,Day]

 signs←13⍴'Capricorn' 'Aquarius' 'Pisces' 'Aries' 'Taurus' 'Gemini' 'Cancer' 'Leo' 'Virgo' 'Libra' 'Scorpio' 'Sa
 dates←119 218 320 419 520 620 722 822 922 1022 1121 1221
 sign←signs⊃⍨1+dates⍸100⊥2↑date
```



Request

Method to Execute: GetSignObject

JSON Data:

```
{month:10, day:31}
```

Response

```
{"day":31,"month":10,"sign":"Scorpio"}
```

C:\Devt\jarvis\Samples\JSON\GetSignObject.dyalog

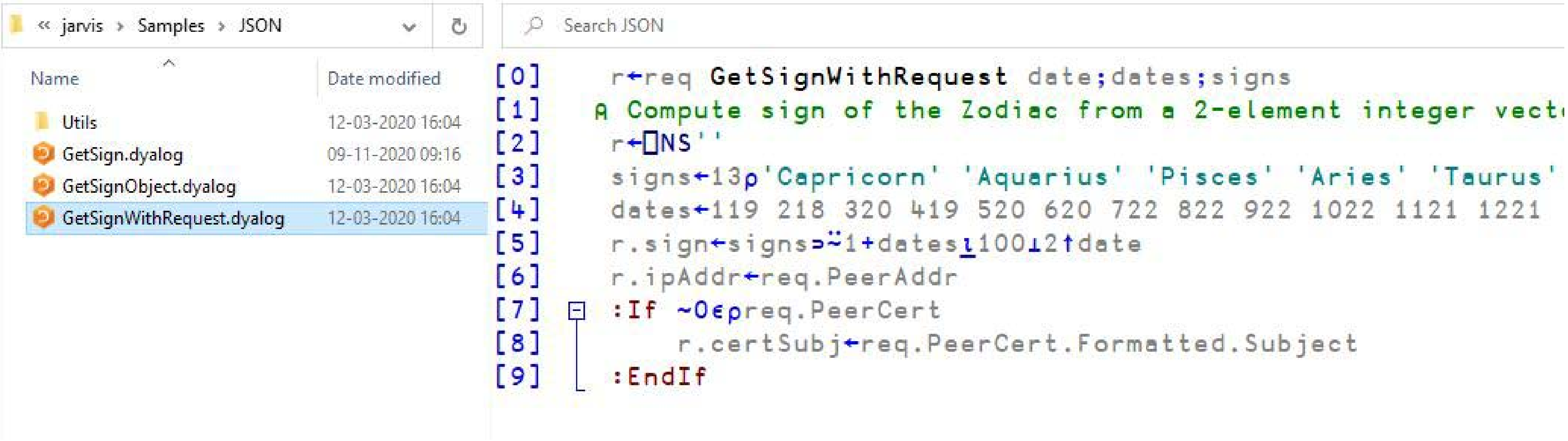
```
ns←GetSignObject ns
⍝ Return a sign object contain month, day (provided as input) and sign

ns.sign←GetSign ns.(month day)
```

Function

Pos: 0/4,1

Jarvis - Sample JSON Application

By default, Jarvis serves a built-in demo application

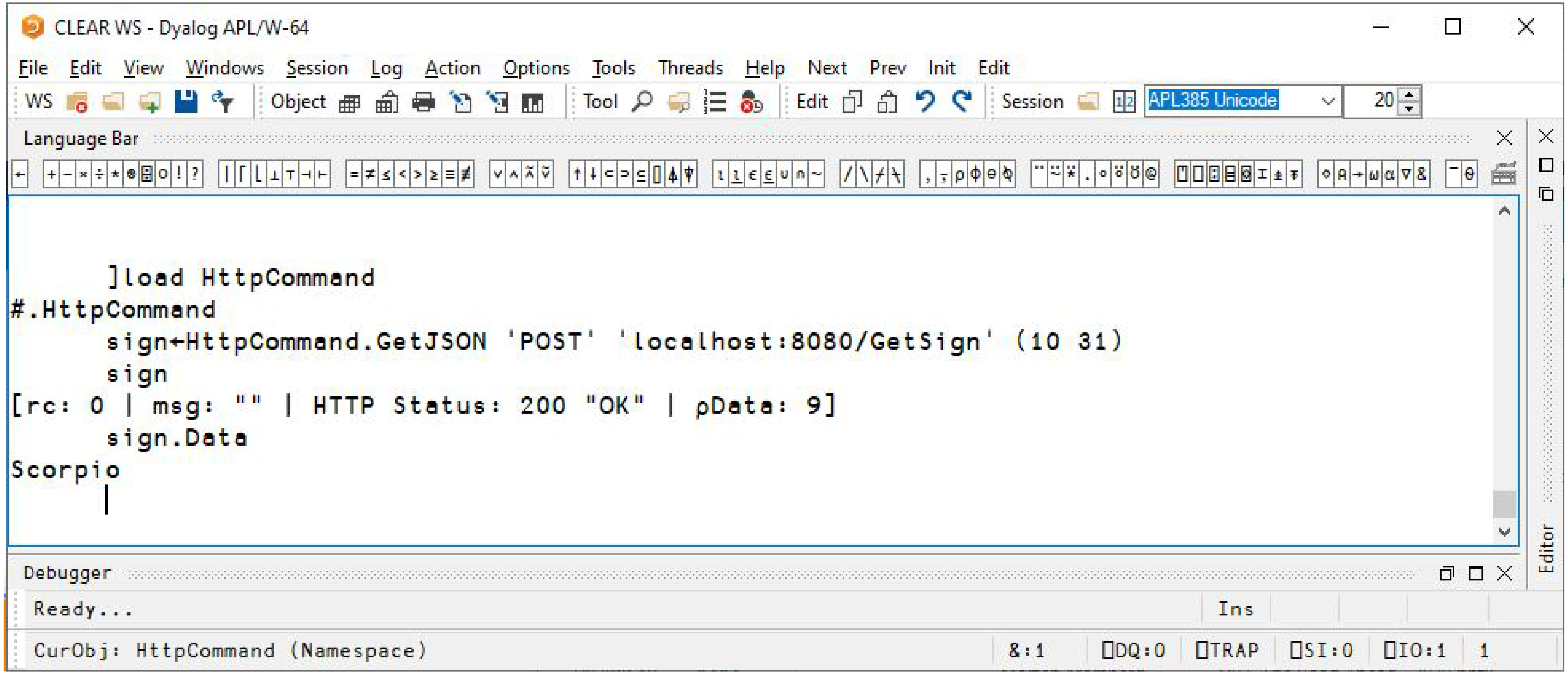


Name	Date modified
Utils	12-03-2020 16:04
GetSign.dyalog	09-11-2020 09:16
GetSignObject.dyalog	12-03-2020 16:04
GetSignWithRequest.dyalog	12-03-2020 16:04

```
    r←req GetSignWithRequest date;dates;signs
  ⍝ Compute sign of the Zodiac from a 2-element integer vect
    r←⎕NS''
    signs←13⍴'Capricorn' 'Aquarius' 'Pisces' 'Aries' 'Taurus'
    dates←119 218 320 419 520 620 722 822 922 1022 1121 1221
    r.sign←signs⊃⍨1+dates⍸100⊥2↑date
    r.ipAddr←req.PeerAddr
    :If ~0∊⍴req.PeerCert
        r.certSubj←req.PeerCert.Formatted.Subject
    :EndIf
```

Demo 2 – Call Jarvis from APL

Call Jarvis service from APL



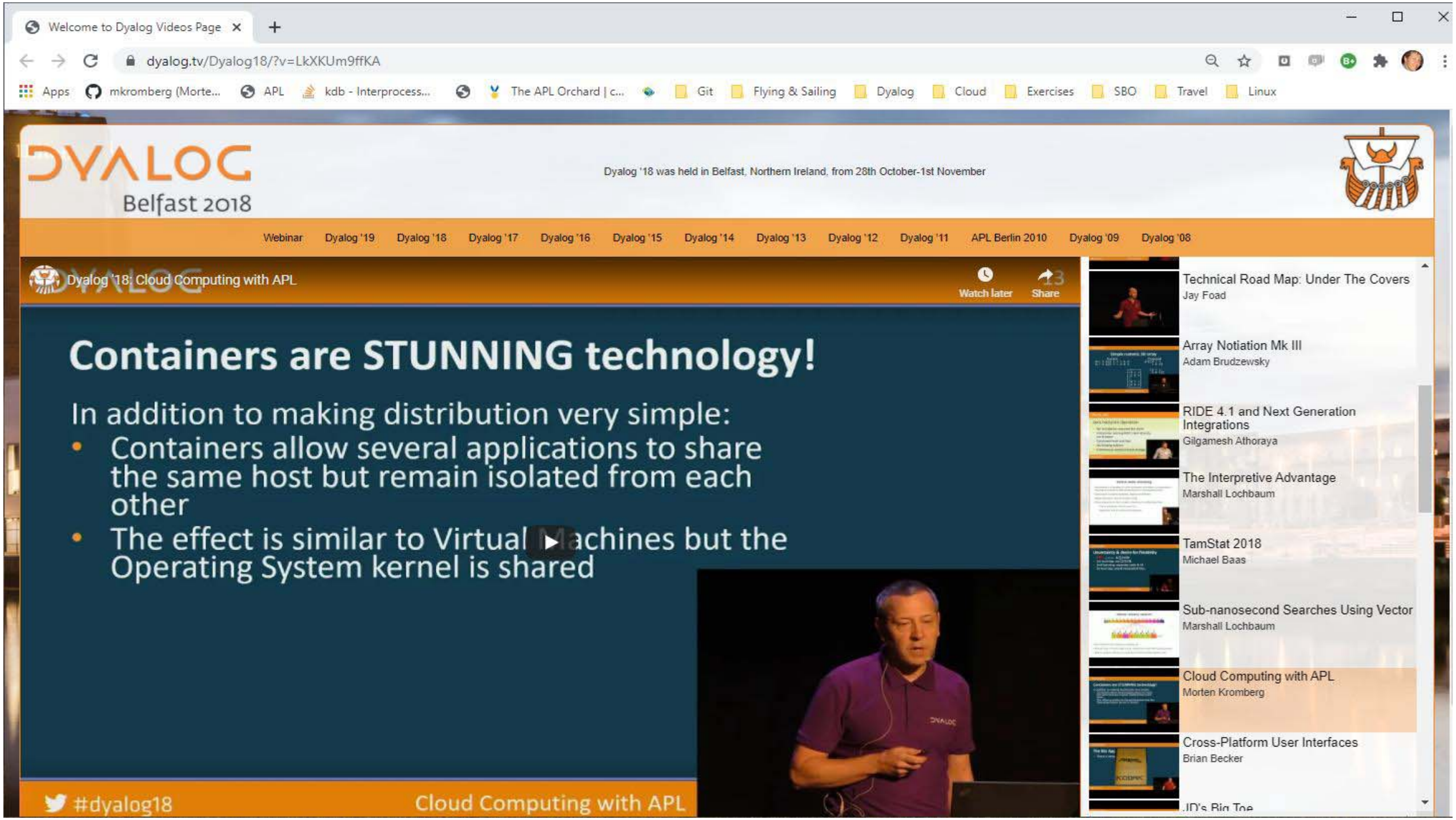
CLEAR WS - Dyalog APL/W-64
File Edit View Windows Session Log Action Options Tools Threads Help Next Prev Init Edit
WS Object Tool Edit Session
APL385 Unicode 20
Language Bar
Debugger
Editor

```
      ]load HttpCommand
#.HttpCommand
      sign←HttpCommand.GetJSON 'POST' 'localhost:8080/GetSign' (10 31)
      sign
[rc: 0 | msg: "" | HTTP Status: 200 "OK" | ⍴Data: 9]
      sign.Data
Scorpio
```

```
Ready...                                   Ins
CurObj: HttpCommand (Namespace)    &:1  ⎕DQ:0  ⎕TRAP  ⎕SI:0  ⎕IO:1  1
```

More details on Docker and APL

- Dyalog'18 (Belfast): Cloud Computing with APL https://dyalog.tv/Dyalog18/?v=LkXKUm9ffKA
- Webinar: Microservices in Dyalog APL https://dyalog.tv/Webinar/?v=KKJMeGAWGaE



Dyalog '18 was held in Belfast, Northern Ireland, from 28th October-1st November

Dyalog '18: Cloud Computing with APL

Containers are STUNNING technology!

In addition to making distribution very simple:

- Containers allow several applications to share the same host but remain isolated from each other
- The effect is similar to Virtual Machines but the Operating System kernel is shared

#dyalog18 Cloud Computing with APL

Technical Road Map: Under The Covers
Jay Foad

Array Notiation Mk III
Adam Brudzewsky

RIDE 4.1 and Next Generation Integrations
Gilgamesh Athoraya

The Interpretive Advantage
Marshall Lochbaum

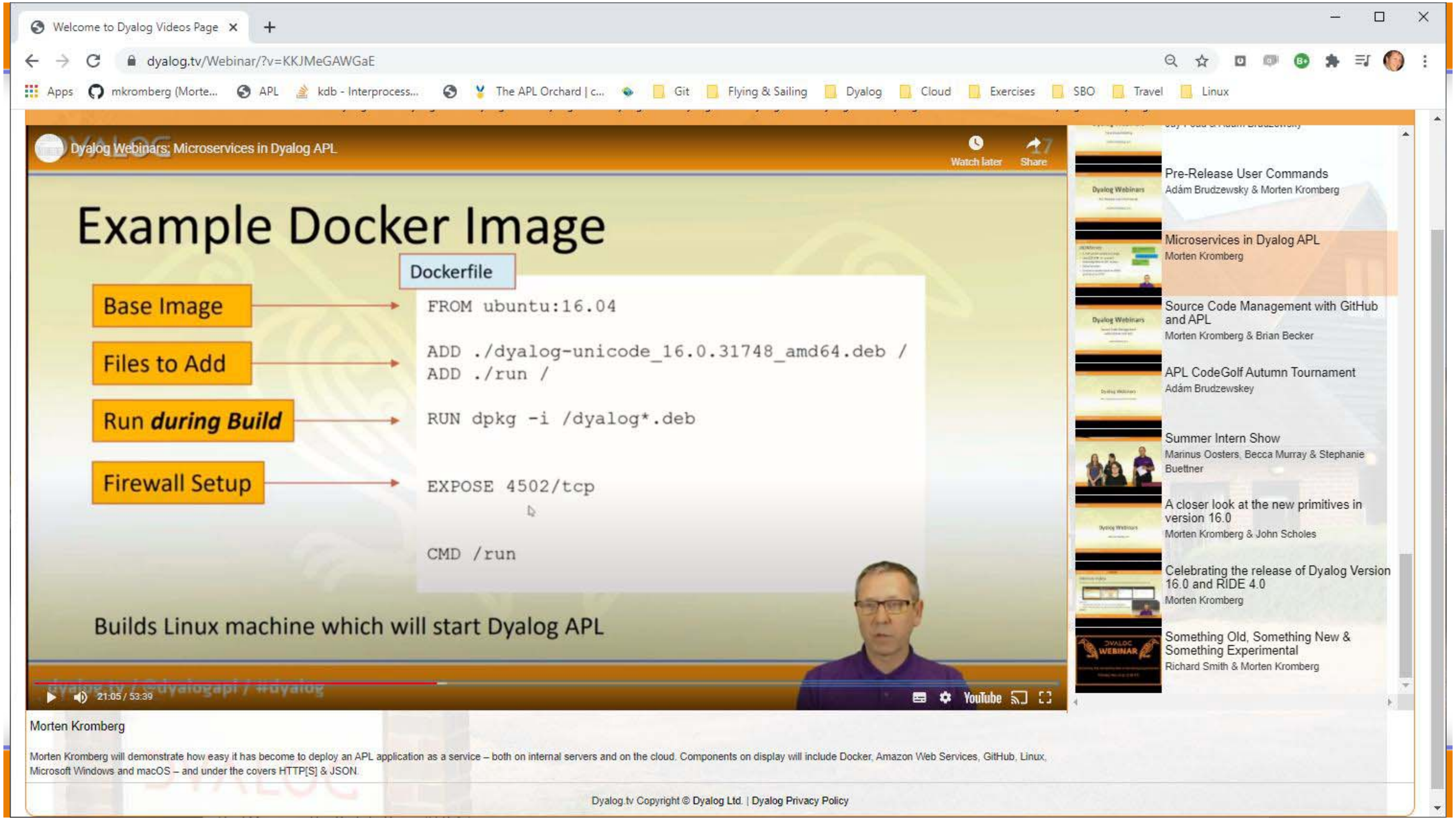
TamStat 2018
Michael Baas

Sub-nanosecond Searches Using Vector
Marshall Lochbaum

Cloud Computing with APL
Morten Kromberg

Cross-Platform User Interfaces
Brian Becker

JD's Big Toe



Example Docker Image

Base Image
Files to Add
Run *during Build*
Firewall Setup

Dockerfile

```
FROM ubuntu:16.04

ADD ./dyalog-unicode_16.0.31748_amd64.deb /
ADD ./run /

RUN dpkg -i /dyalog*.deb


EXPOSE 4502/tcp


CMD /run
```

Builds Linux machine which will start Dyalog APL

Pre-Release User Commands
Adám Brudzewsky & Morten Kromberg

Microservices in Dyalog APL
Morten Kromberg

Source Code Management with GitHub and APL
Morten Kromberg & Brian Becker

APL CodeGolf Autumn Tournament
Adám Brudzewskey

Summer Intern Show
Marinus Oosters, Becca Murray & Stephanie Buettner

A closer look at the new primitives in version 16.0
Morten Kromberg & John Scholes

Celebrating the release of Dyalog Version 16.0 and RIDE 4.0
Morten Kromberg

Something Old, Something New & Something Experimental
Richard Smith & Morten Kromberg

Morten Kromberg

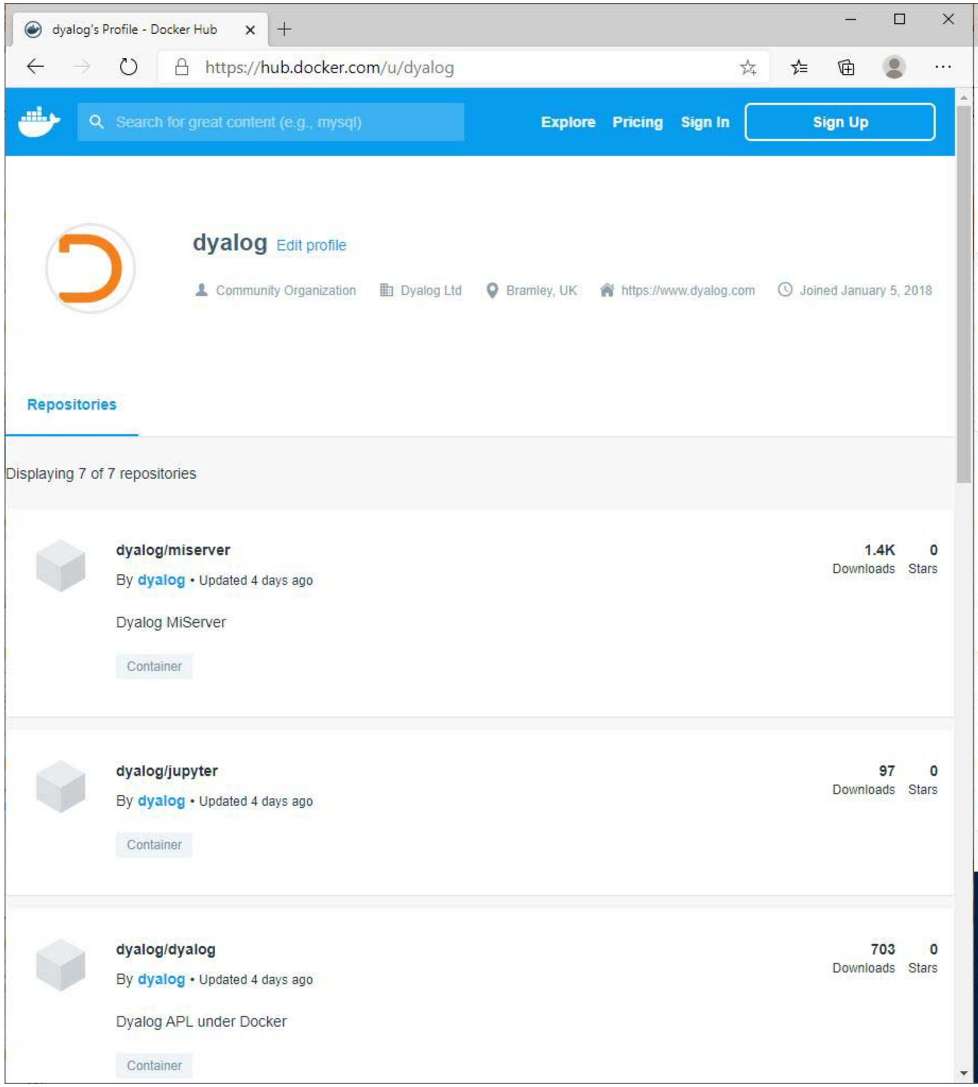
Morten Kromberg will demonstrate how easy it has become to deploy an APL application as a service – both on internal servers and on the cloud. Components on display will include Docker, Amazon Web Services, GitHub, Linux, Microsoft Windows and macOS – and under the covers HTTP[S] & JSON.

Dyalog.tv Copyright © Dyalog Ltd. | Dyalog Privacy Policy

Dyalog's Public Containers

At https://hub.docker.com/u/dyalog

Image Name	Description
dyalog/dyalog	Minimal Linux with Dyalog APL (currently debian:buster-slim)
dyalog/jarvis	dyalog/dyalog with Jarvis installed
dyalog/miserver	... with MiServer installed
dyalog/jupyter	For Dyalog-backed Jupyter notebooks
dyalog/jsonserver	Use dyalog/jarvis instead


dyalog's Profile - Docker Hub
https://hub.docker.com/u/dyalog
Search for great content (e.g., mysql)
Explore
Pricing
Sign In
Sign Up

dyalog Edit profile

Community Organization · Dyalog Ltd · Bramley, UK · https://www.dyalog.com · Joined January 5, 2018

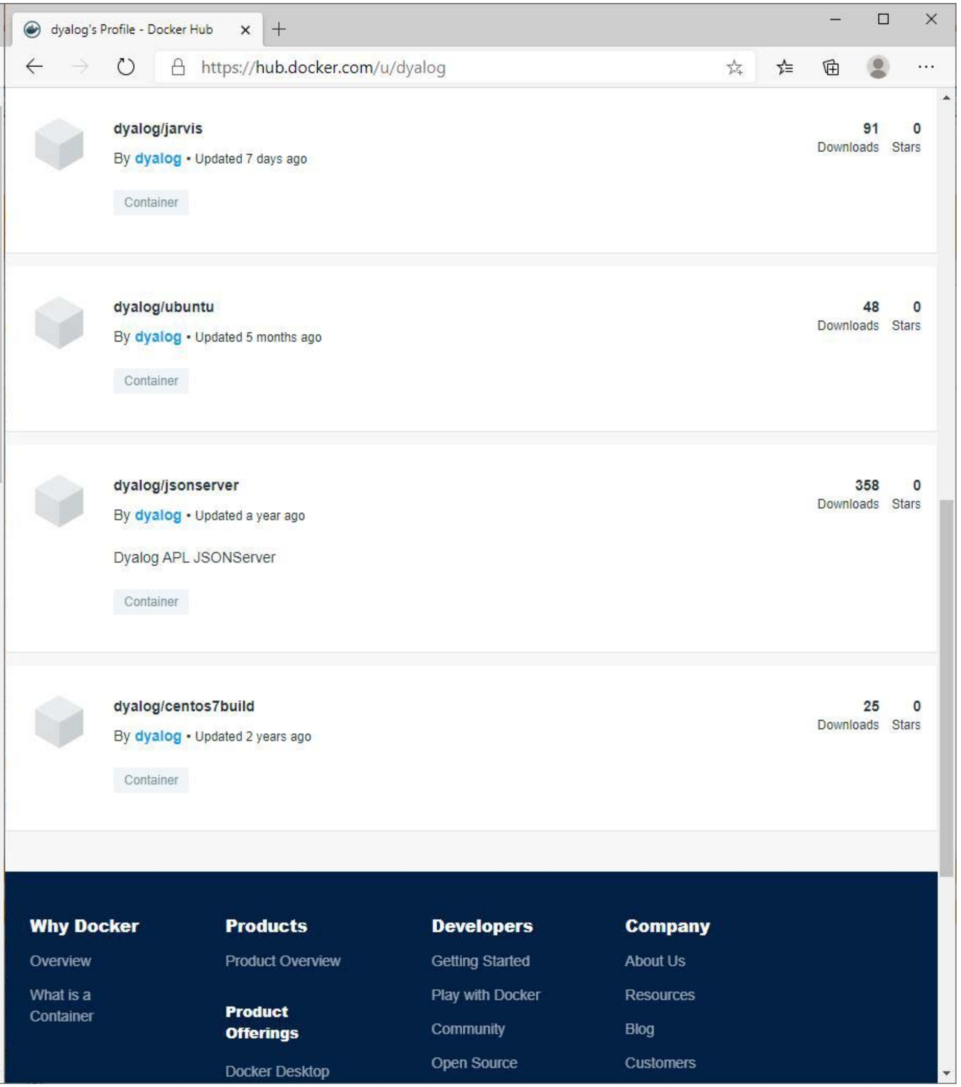
Repositories

Displaying 7 of 7 repositories

dyalog/miserver
By dyalog • Updated 4 days ago
Dyalog MiServer
Container
1.4K Downloads · 0 Stars

dyalog/jupyter
By dyalog • Updated 4 days ago
Container
97 Downloads · 0 Stars

dyalog/dyalog
By dyalog • Updated 4 days ago
Dyalog APL under Docker
Container
703 Downloads · 0 Stars

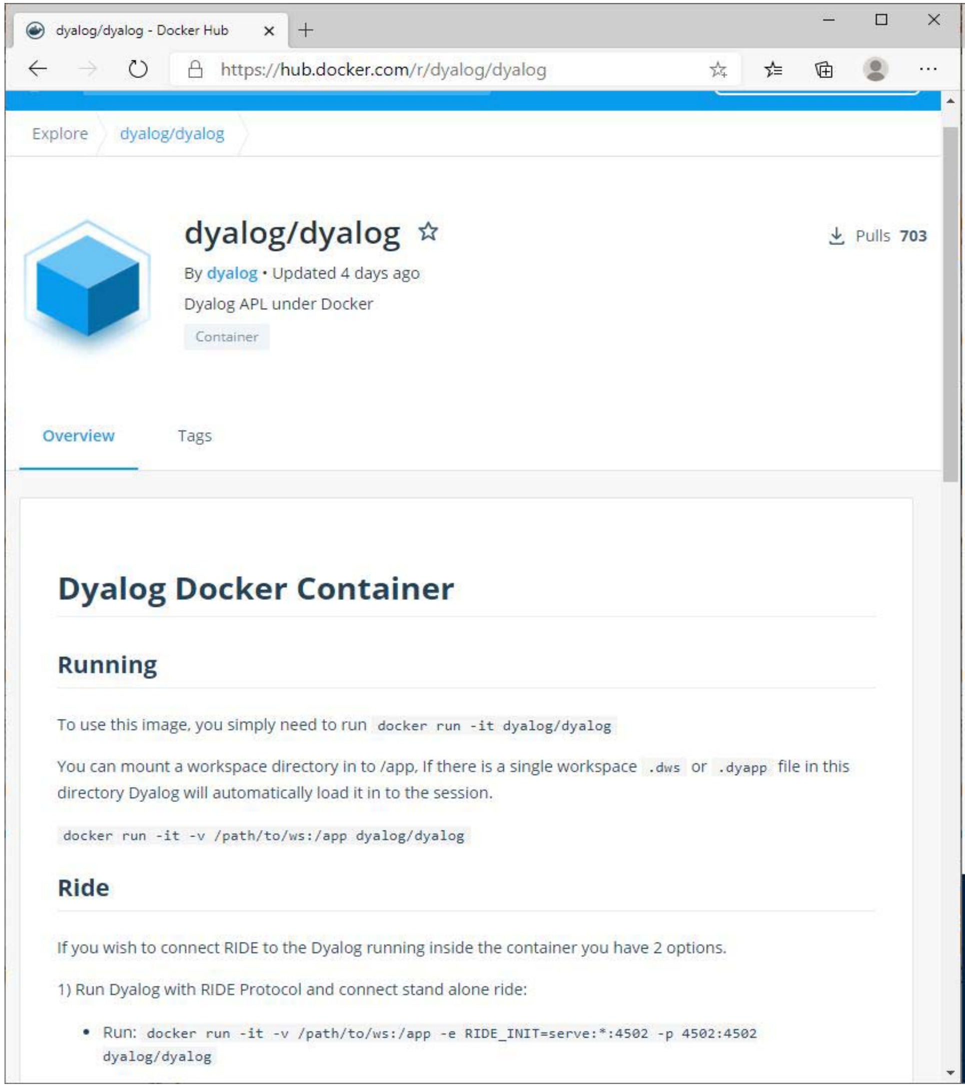

dyalog's Profile - Docker Hub
https://hub.docker.com/u/dyalog

dyalog/jarvis
By dyalog • Updated 7 days ago
Container
91 Downloads · 0 Stars

dyalog/ubuntu
By dyalog • Updated 5 months ago
Container
48 Downloads · 0 Stars

dyalog/jsonserver
By dyalog • Updated a year ago
Dyalog APL JSONServer
Container
358 Downloads · 0 Stars

dyalog/centos7build
By dyalog • Updated 2 years ago
Container
25 Downloads · 0 Stars



dyalog/dyalog - Docker Hub
https://hub.docker.com/r/dyalog/dyalog
Explore
dyalog/dyalog

dyalog/dyalog ☆

Pulls 703

By dyalog • Updated 4 days ago

Dyalog APL under Docker

Container

Dyalog Docker Container

Running

To use this image, you simply need to run `docker run -it dyalog/dyalog`

You can mount a workspace directory in to /app, If there is a single workspace `.dws` or `.dyapp` file in this directory Dyalog will automatically load it in to the session.

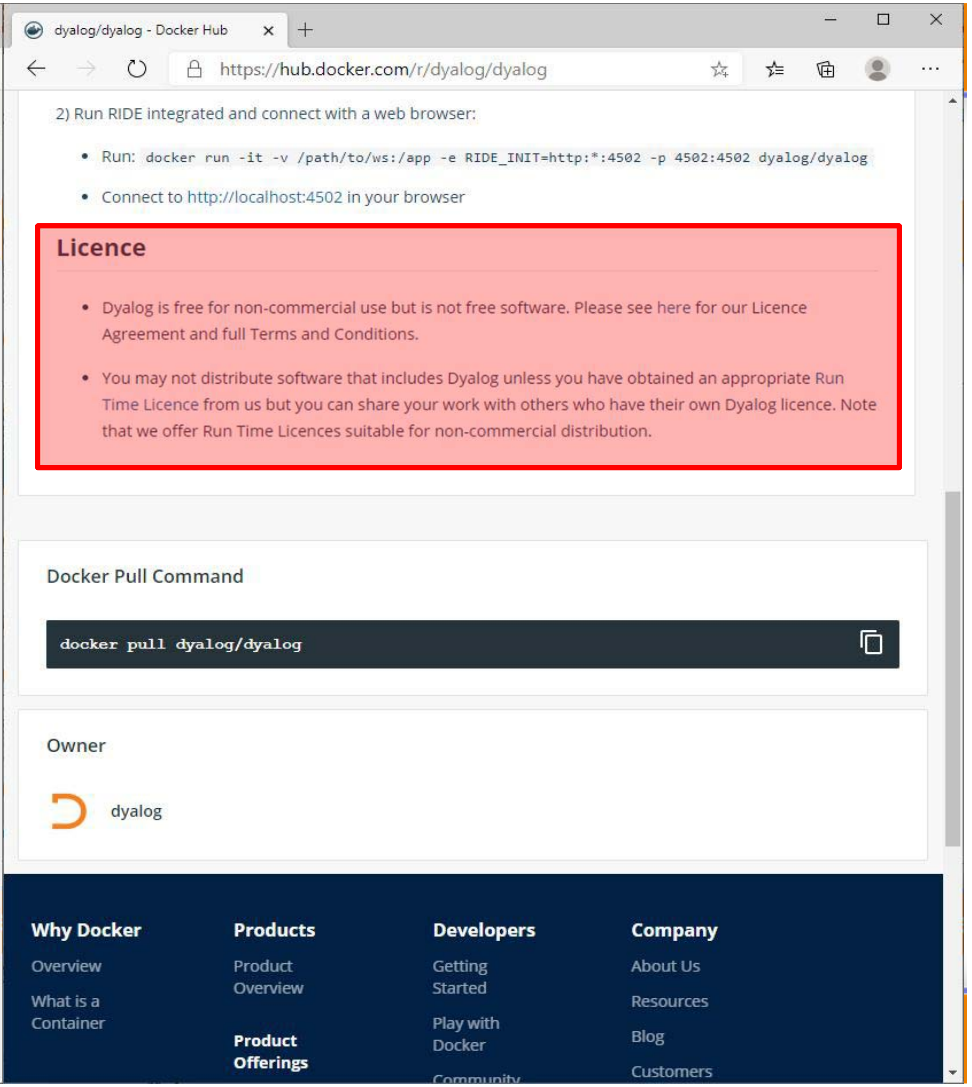
```
docker run -it -v /path/to/ws:/app dyalog/dyalog
```

Ride

If you wish to connect RIDE to the Dyalog running inside the container you have 2 options.

1) Run Dyalog with RIDE Protocol and connect stand alone ride:

- Run: `docker run -it -v /path/to/ws:/app -e RIDE_INIT=serve:*:4502 -p 4502:4502 dyalog/dyalog`



dyalog/dyalog - Docker Hub
https://hub.docker.com/r/dyalog/dyalog

2) Run RIDE integrated and connect with a web browser:

- Run: `docker run -it -v /path/to/ws:/app -e RIDE_INIT=http:*:4502 -p 4502:4502 dyalog/dyalog`
- Connect to http://localhost:4502 in your browser

Licence

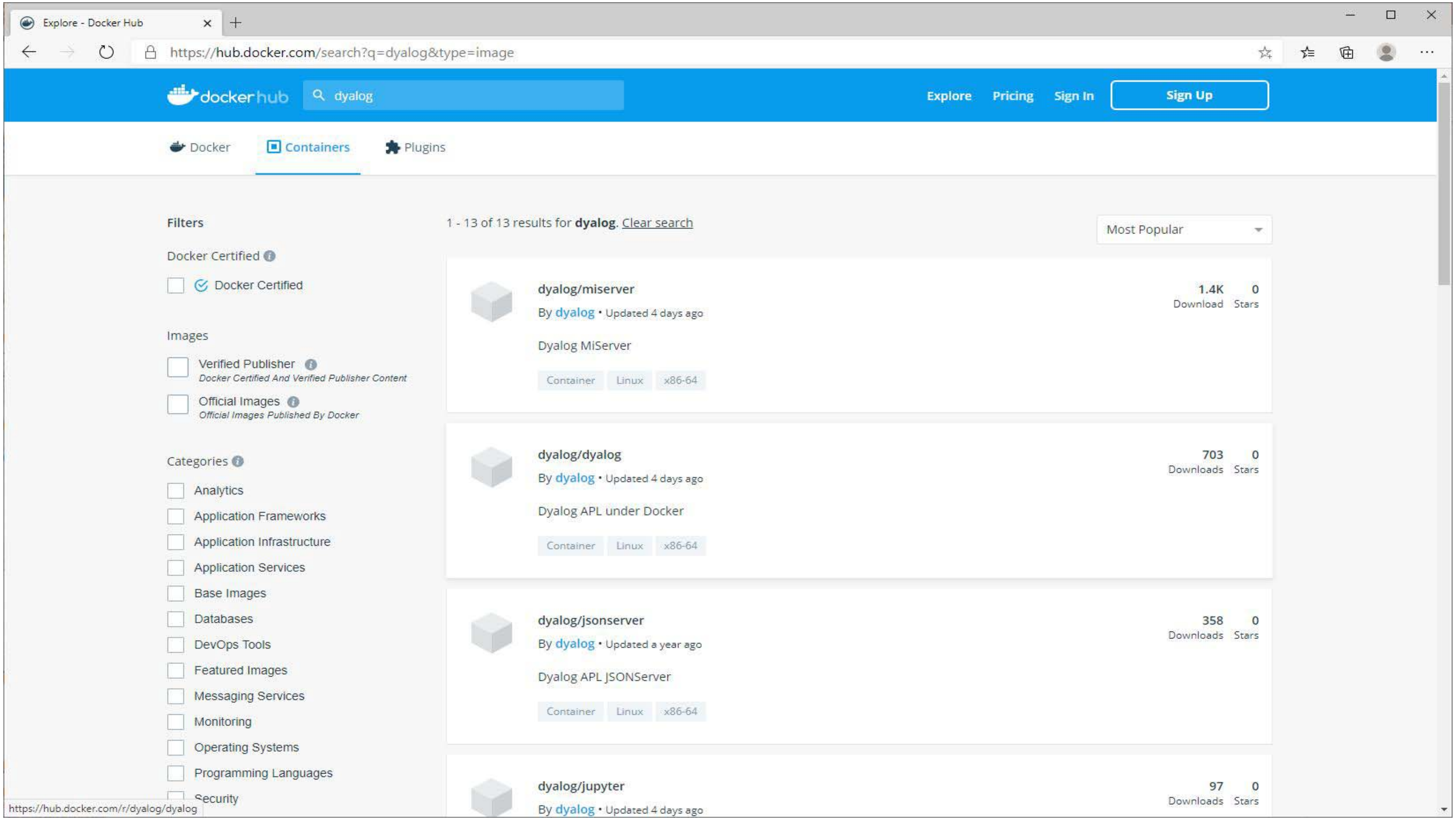
- Dyalog is free for non-commercial use but is not free software. Please see here for our Licence Agreement and full Terms and Conditions.
- You may not distribute software that includes Dyalog unless you have obtained an appropriate Run Time Licence from us but you can share your work with others who have their own Dyalog licence. Note that we offer Run Time Licences suitable for non-commercial distribution.

Docker Pull Command

```
docker pull dyalog/dyalog
```

Owner

dyalog



Filters

Docker Certified

□ Docker Certified

Images

□ Verified Publisher
Docker Certified And Verified Publisher Content

□ Official Images
Official Images Published By Docker

Categories

□ Analytics
□ Application Frameworks
□ Application Infrastructure
□ Application Services
□ Base Images
□ Databases
□ DevOps Tools
□ Featured Images
□ Messaging Services
□ Monitoring
□ Operating Systems
□ Programming Languages
□ Security

1 - 13 of 13 results for **dyalog**. Clear search

Most Popular

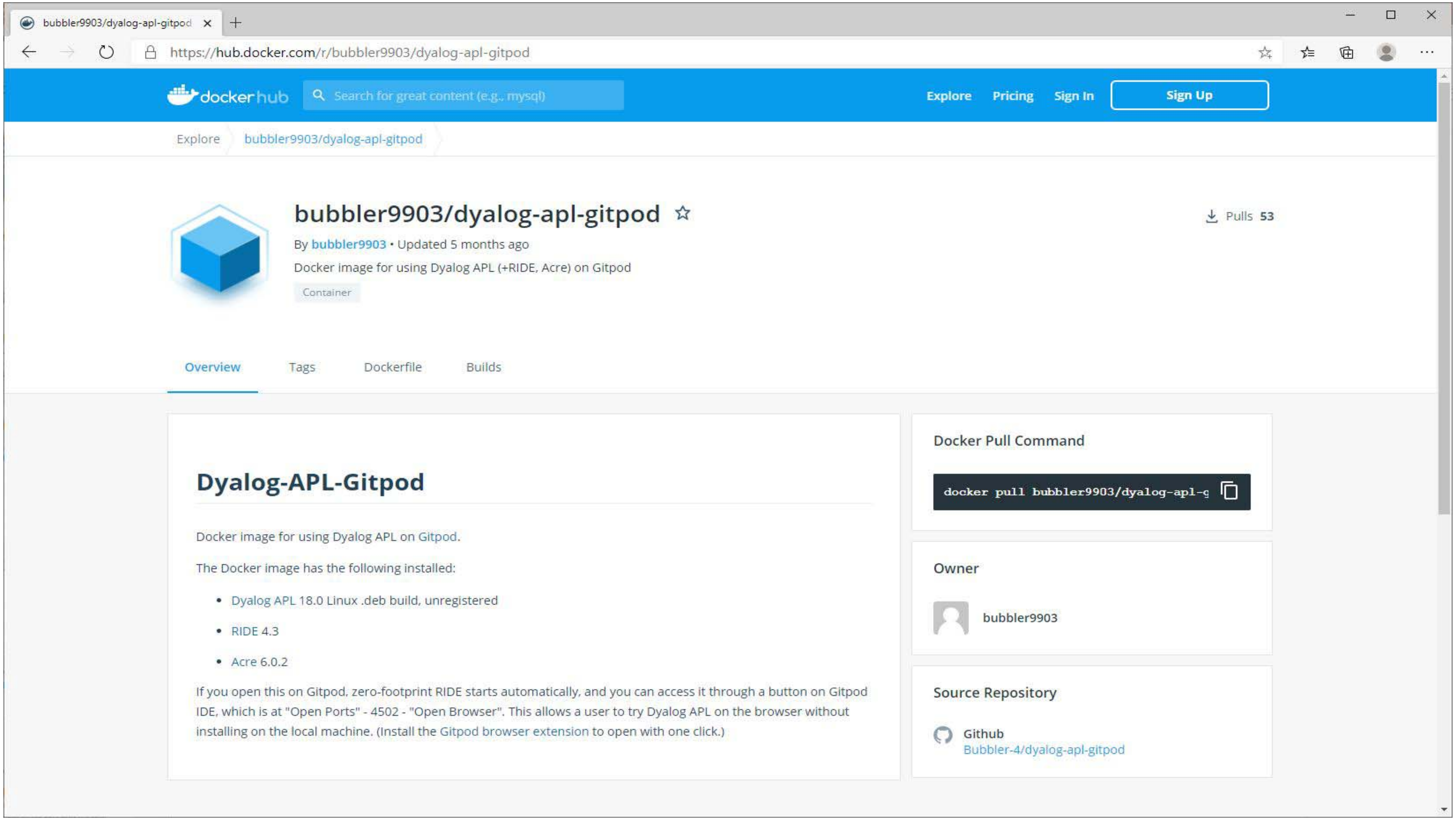
dyalog/miserver
By dyalog • Updated 4 days ago
Dyalog MiServer
Container Linux x86-64
1.4K Download 0 Stars

dyalog/dyalog
By dyalog • Updated 4 days ago
Dyalog APL under Docker
Container Linux x86-64
703 Downloads 0 Stars

dyalog/jsonserver
By dyalog • Updated a year ago
Dyalog APL JSONServer
Container Linux x86-64
358 Downloads 0 Stars

dyalog/jupyter
By dyalog • Updated 4 days ago
97 Downloads 0 Stars

https://hub.docker.com/r/dyalog/dyalog



bubbler9903/dyalog-apl-gitpod
https://hub.docker.com/r/bubbler9903/dyalog-apl-gitpod
docker hub
Search for great content (e.g., mysql)
Explore
Pricing
Sign In
Sign Up
Explore
bubbler9903/dyalog-apl-gitpod

bubbler9903/dyalog-apl-gitpod

By bubbler9903 • Updated 5 months ago

Docker image for using Dyalog APL (+RIDE, Acre) on Gitpod

Container

Pulls 53

Overview
Tags
Dockerfile
Builds

Dyalog-APL-Gitpod

Docker image for using Dyalog APL on Gitpod.

The Docker image has the following installed:

- Dyalog APL 18.0 Linux .deb build, unregistered
- RIDE 4.3
- Acre 6.0.2

If you open this on Gitpod, zero-footprint RIDE starts automatically, and you can access it through a button on Gitpod IDE, which is at "Open Ports" - 4502 - "Open Browser". This allows a user to try Dyalog APL on the browser without installing on the local machine. (Install the Gitpod browser extension to open with one click.)

Docker Pull Command

```
docker pull bubbler9903/dyalog-apl-g
```

Owner

bubbler9903

Source Repository

Github
Bubbler-4/dyalog-apl-gitpod

Boot Folders

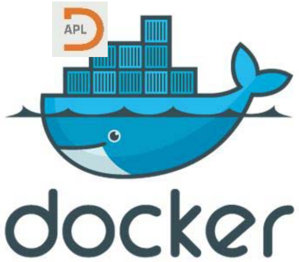
Each of our public containers has a default folder which it will look for:

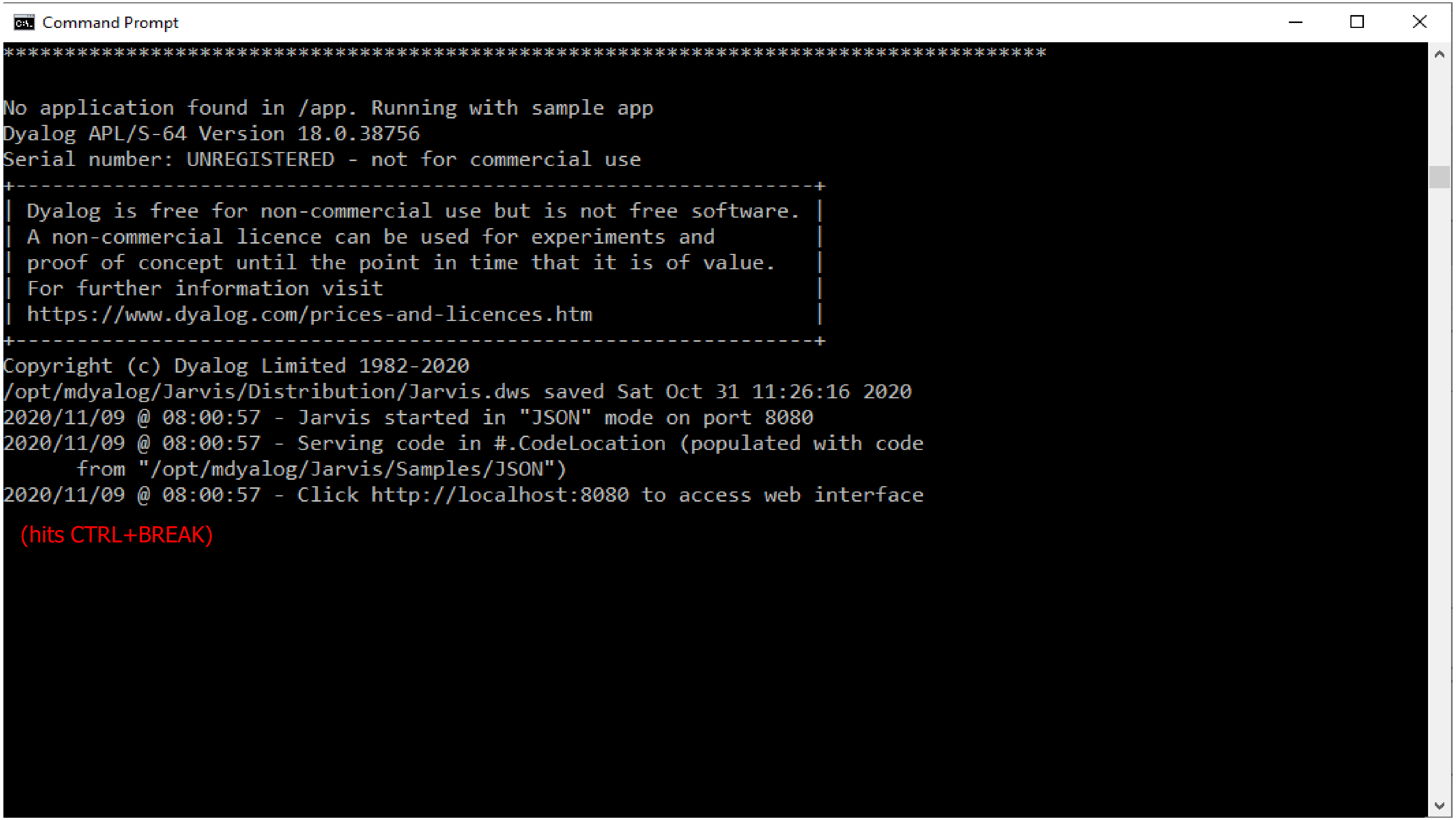
Image Name	Description	Boot Folder
dyalog/dyalog	Minimal Linux with Dyalog APL	/app
dyalog/jarvis	dyalog/dyalog with Jarvis installed	/code
dyalog/jsonserver	Use dyalog/jarvis instead	/code
dyalog/miserver	... with MiServer installed	/misite
dyalog/jupyter	For Dyalog-backed Jupyter notebooks	/app

```
docker run -it -v /my/app/folder:/code dyalog/jarvis
```

Stopping a Container

- Implement some kind of ShutDown method in your service
  - The container will shut down when the initial process started by CMD terminates
- Connect RIDE if you can and type)OFF
- or ...





```
************************************************************************************

No application found in /app. Running with sample app
Dyalog APL/S-64 Version 18.0.38756
Serial number: UNREGISTERED - not for commercial use
+-----------------------------------------------------------------+
| Dyalog is free for non-commercial use but is not free software. |
| A non-commercial licence can be used for experiments and        |
| proof of concept until the point in time that it is of value.   |
| For further information visit                                   |
| https://www.dyalog.com/prices-and-licences.htm                  |
+-----------------------------------------------------------------+
Copyright (c) Dyalog Limited 1982-2020
/opt/mdyalog/Jarvis/Distribution/Jarvis.dws saved Sat Oct 31 11:26:16 2020
2020/11/09 @ 08:00:57 - Jarvis started in "JSON" mode on port 8080
2020/11/09 @ 08:00:57 - Serving code in #.CodeLocation (populated with code
      from "/opt/mdyalog/Jarvis/Samples/JSON")
2020/11/09 @ 08:00:57 - Click http://localhost:8080 to access web interface
```

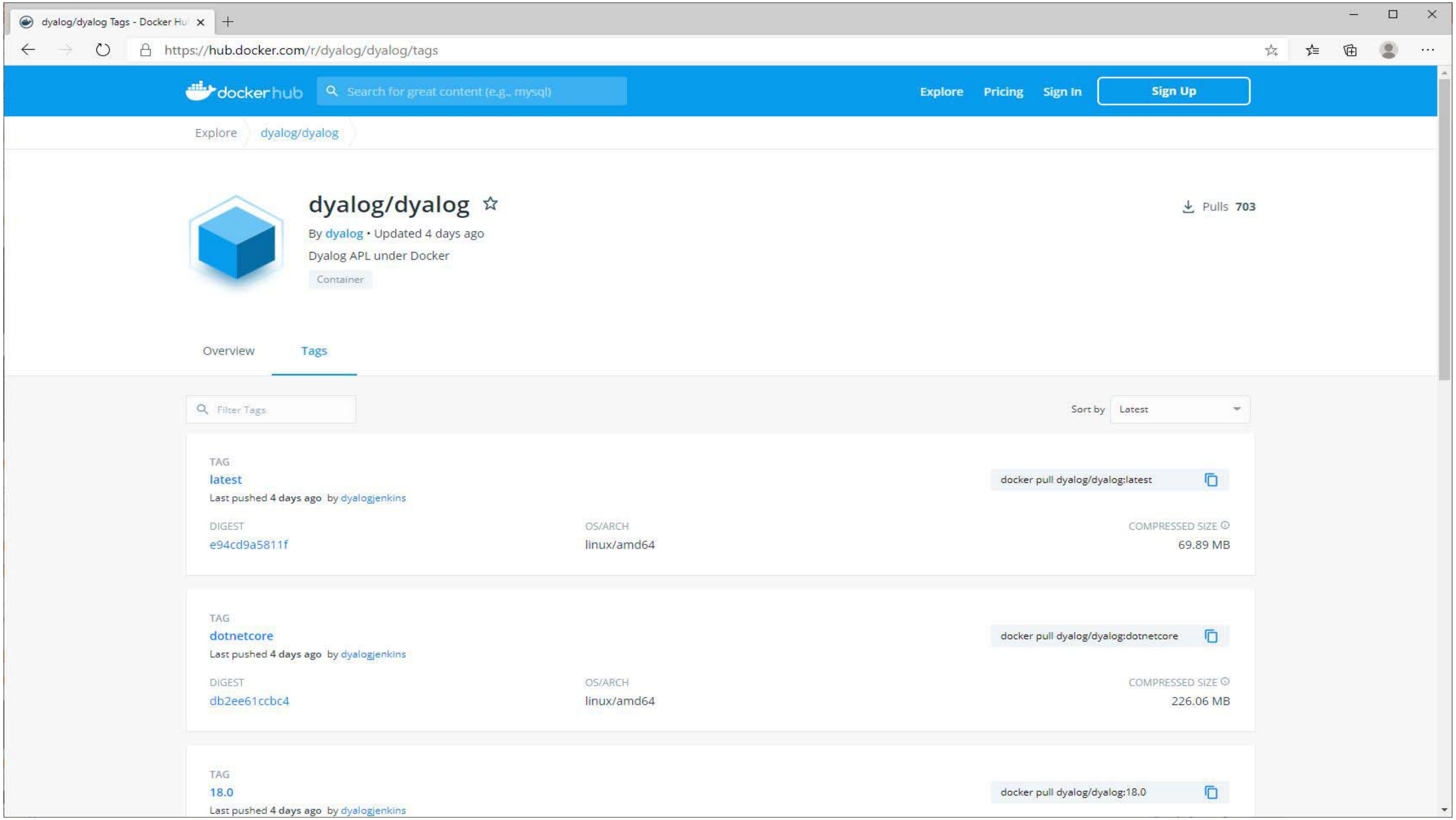
(hits CTRL+BREAK)

Tags

- Images may have several tags, identifying variants
- dyalog/dyalog has the following tags
  - latest
  - 17.1
  - 18.0
  - dotnetcore

(Will be renamed 18.0-dotnetcore, 19.0-dotnetcore)

```
docker run -it -e RIDE_INIT=HTTP:*:4502
               -p 4502:4502 dyalog/dyalog:17.1
```



https://hub.docker.com/r/dyalog/dyalog/tags

dyalog/dyalog

By dyalog • Updated 4 days ago

Dyalog APL under Docker

Container

Pulls 703

TAG
latest
Last pushed 4 days ago by dyalogjenkins

`docker pull dyalog/dyalog:latest`

DIGEST	OS/ARCH	COMPRESSED SIZE
e94cd9a5811f	linux/amd64	69.89 MB

TAG
dotnetcore
Last pushed 4 days ago by dyalogjenkins

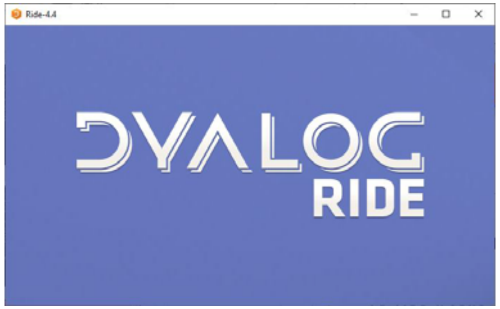
`docker pull dyalog/dyalog:dotnetcore`

DIGEST	OS/ARCH	COMPRESSED SIZE
db2ee61ccbc4	linux/amd64	226.06 MB

TAG
18.0
Last pushed 4 days ago by dyalogjenkins

`docker pull dyalog/dyalog:18.0`

More about Debugging



- To be able to use RIDE for debugging, add

```
-e RIDE_INIT=SERVE:*:4502 -p 4502:4502
```

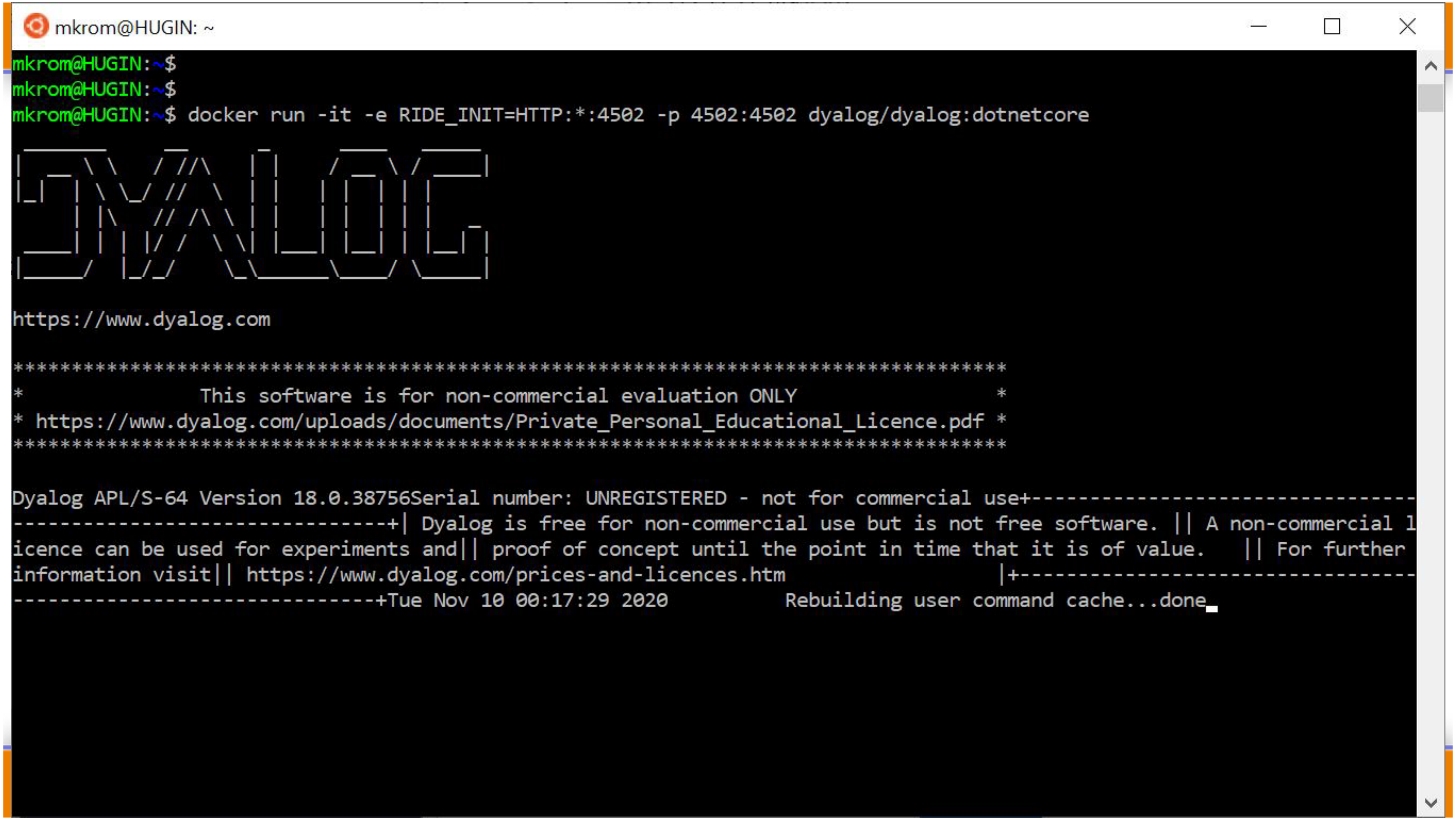
- To use "Zero Footprint" RIDE via a browser, use HTTP mode:

```
-e RIDE_INIT=HTTP:*:4502 -p 4502:4502
```

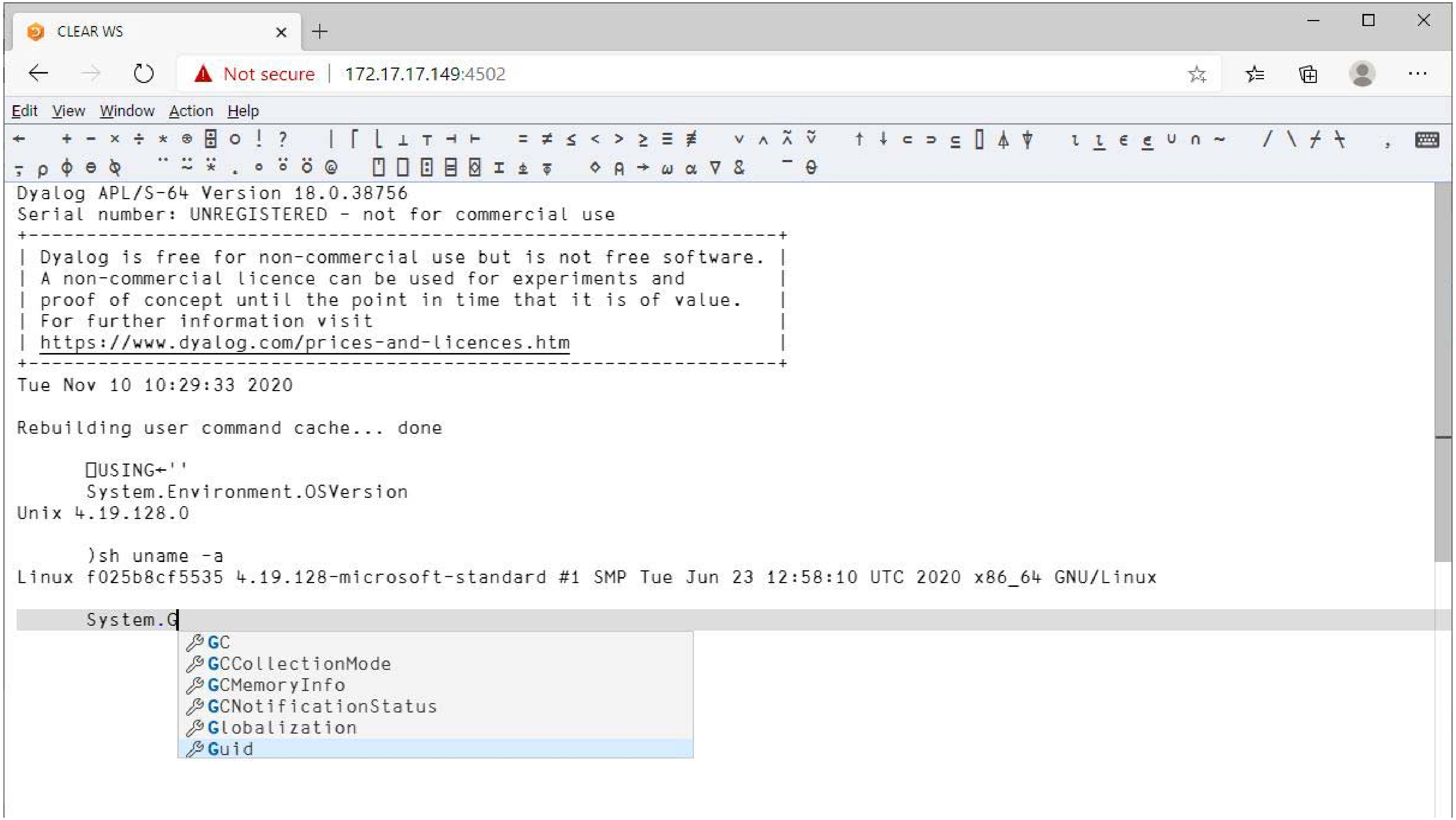
Live demo time...

Demo 3

- Start APL, loading and running a function
- Debug using pre-installed RIDE
- Repeat using "Zero Footprint" RIDE



```
mkrom@HUGIN:~$
mkrom@HUGIN:~$
mkrom@HUGIN:~$ docker run -it -e RIDE_INIT=HTTP:*:4502 -p 4502:4502 dyalog/dyalog:dotnetcore
 _______     __      _      ____   _____
|  __ \ \   / //\   | |    / __ \ / ____|
|_|  | \ \_/ //  \  | |   | |  | | |
     | |\   // /\ \ | |   | |  | | |   _
 ____| | | |/ /  \ \| |___| |__| | |__| |
|_____/  |_/_/    \_\______\____/ \_____|

https://www.dyalog.com

*************************************************************************************
*               This software is for non-commercial evaluation ONLY                 *
* https://www.dyalog.com/uploads/documents/Private_Personal_Educational_Licence.pdf *
*************************************************************************************

Dyalog APL/S-64 Version 18.0.38756Serial number: UNREGISTERED - not for commercial use+---------------------------------
--------------------------------+| Dyalog is free for non-commercial use but is not free software. || A non-commercial l
icence can be used for experiments and|| proof of concept until the point in time that it is of value.   || For further
information visit|| https://www.dyalog.com/prices-and-licences.htm                  |+----------------------------------
-------------------------------+Tue Nov 10 00:17:29 2020          Rebuilding user command cache...done
```

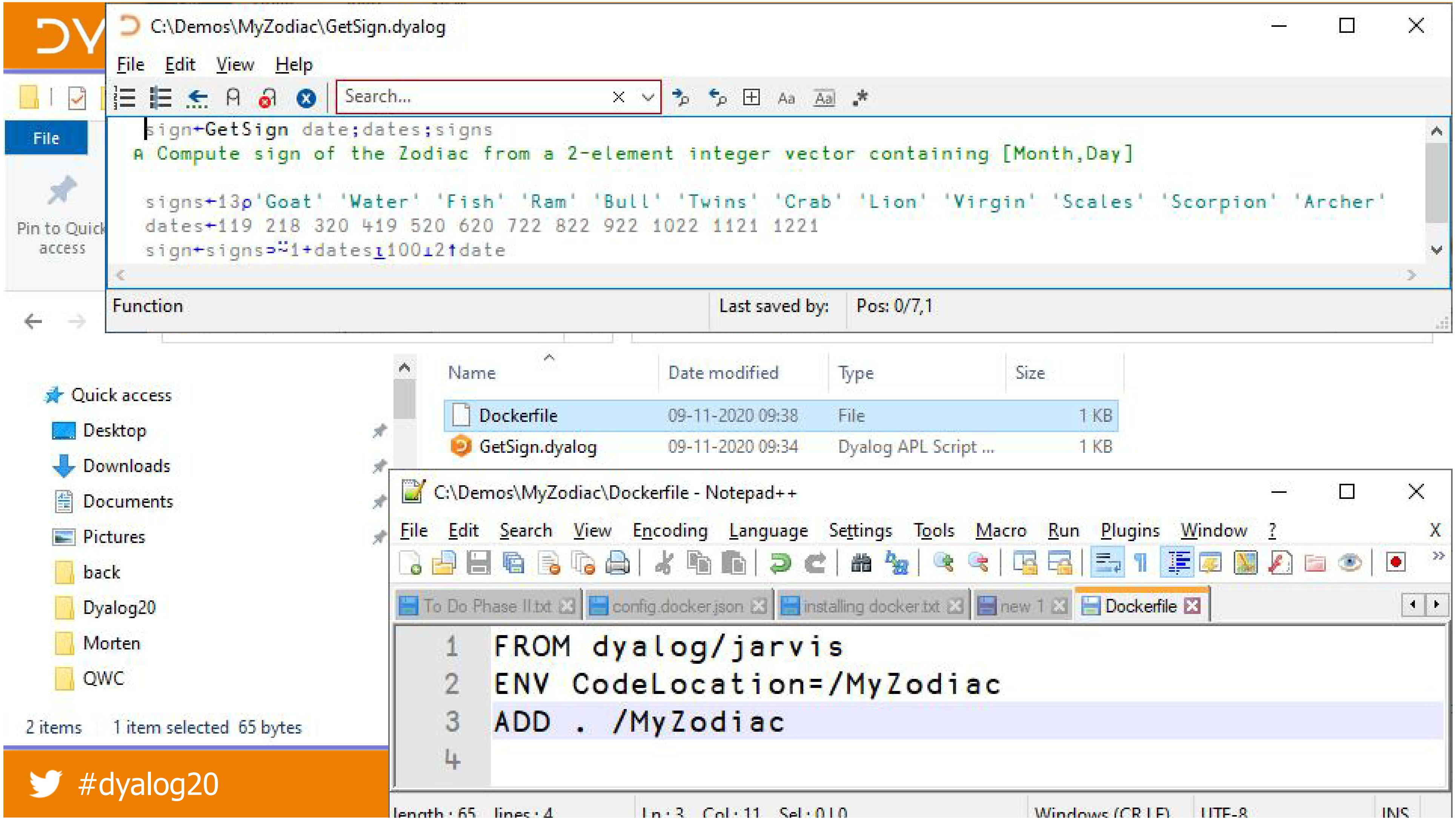


```
Dyalog APL/S-64 Version 18.0.38756
Serial number: UNREGISTERED - not for commercial use
+-----------------------------------------------------------------+
| Dyalog is free for non-commercial use but is not free software. |
| A non-commercial licence can be used for experiments and        |
| proof of concept until the point in time that it is of value.   |
| For further information visit                                   |
| https://www.dyalog.com/prices-and-licences.htm                  |
+-----------------------------------------------------------------+
Tue Nov 10 10:29:33 2020

Rebuilding user command cache... done

      ⎕USING←''
      System.Environment.OSVersion
Unix 4.19.128.0

      )sh uname -a
Linux f025b8cf5535 4.19.128-microsoft-standard #1 SMP Tue Jun 23 12:58:10 UTC 2020 x86_64 GNU/Linux

      System.G
```

Extending Dyalog's Containers

You can make new containers which `extend` Dyalog's containers by adding:

- Your application code
- Your data
    - ... except it usually makes more sense to map data in at runtime



C:\Demos\MyZodiac\GetSign.dyalog

```
 sign←GetSign date;dates;signs
⍝ Compute sign of the Zodiac from a 2-element integer vector containing [Month,Day]

 signs←13⍴'Goat' 'Water' 'Fish' 'Ram' 'Bull' 'Twins' 'Crab' 'Lion' 'Virgin' 'Scales' 'Scorpion' 'Archer'
 dates←119 218 320 419 520 620 722 822 922 1022 1121 1221
 sign←signs⊃⍨1+dates⍸100⊥2↑date
```

Function | Last saved by: | Pos: 0/7,1

Name	Date modified	Type	Size
Dockerfile	09-11-2020 09:38	File	1 KB
GetSign.dyalog	09-11-2020 09:34	Dyalog APL Script ...	1 KB

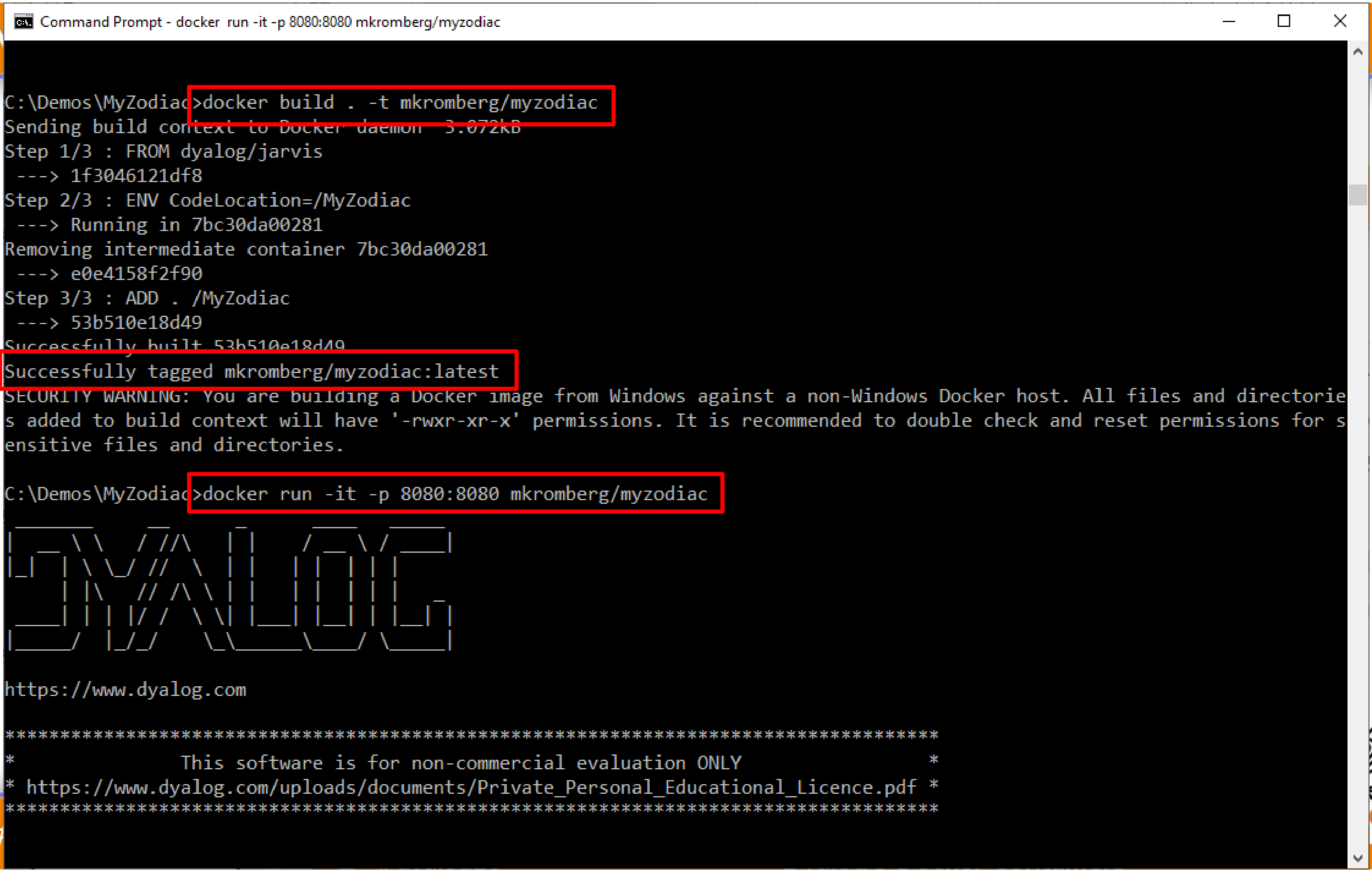
2 items 1 item selected 65 bytes

C:\Demos\MyZodiac\Dockerfile - Notepad++

```
FROM dyalog/jarvis
ENV CodeLocation=/MyZodiac
ADD . /MyZodiac
```

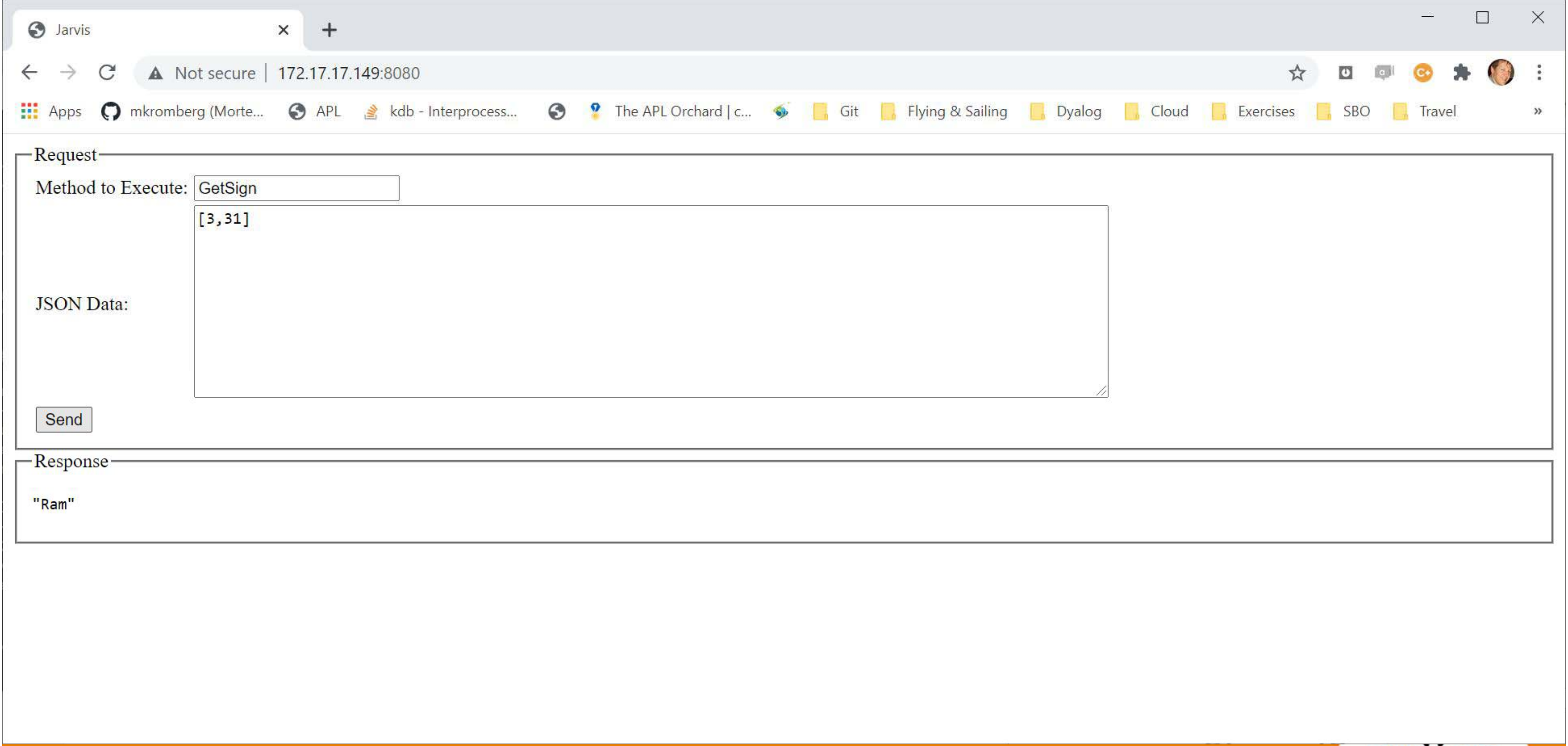
length : 65 lines : 4 Ln : 3 Col : 11 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

#dyalog20



Command Prompt - docker run -it -p 8080:8080 mkromberg/myzodiac

```
C:\Demos\MyZodiac>docker build . -t mkromberg/myzodiac
Sending build context to Docker daemon  3.072kB
Step 1/3 : FROM dyalog/jarvis
 ---> 1f3046121df8
Step 2/3 : ENV CodeLocation=/MyZodiac
 ---> Running in 7bc30da00281
Removing intermediate container 7bc30da00281
 ---> e0e4158f2f90
Step 3/3 : ADD . /MyZodiac
 ---> 53b510e18d49
Successfully built 53b510e18d49
Successfully tagged mkromberg/myzodiac:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and directorie
s added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permissions for s
ensitive files and directories.

C:\Demos\MyZodiac>docker run -it -p 8080:8080 mkromberg/myzodiac
 _______     __      _      ____   _____
|  __ \ \   / //\   | |    / __ \ / ____|
|_|  | \ \_/ //  \  | |   | |  | | | |
     | |\   // /\ \ | |   | |  | | |  _
 ____| | | |/ /  \ \| |___| |__| | | |__| |
|_____/  |_/_/    \_\______\____/ \_____|

https://www.dyalog.com

*************************************************************************************
*                This software is for non-commercial evaluation ONLY                *
* https://www.dyalog.com/uploads/documents/Private_Personal_Educational_Licence.pdf *
*************************************************************************************
```



Request

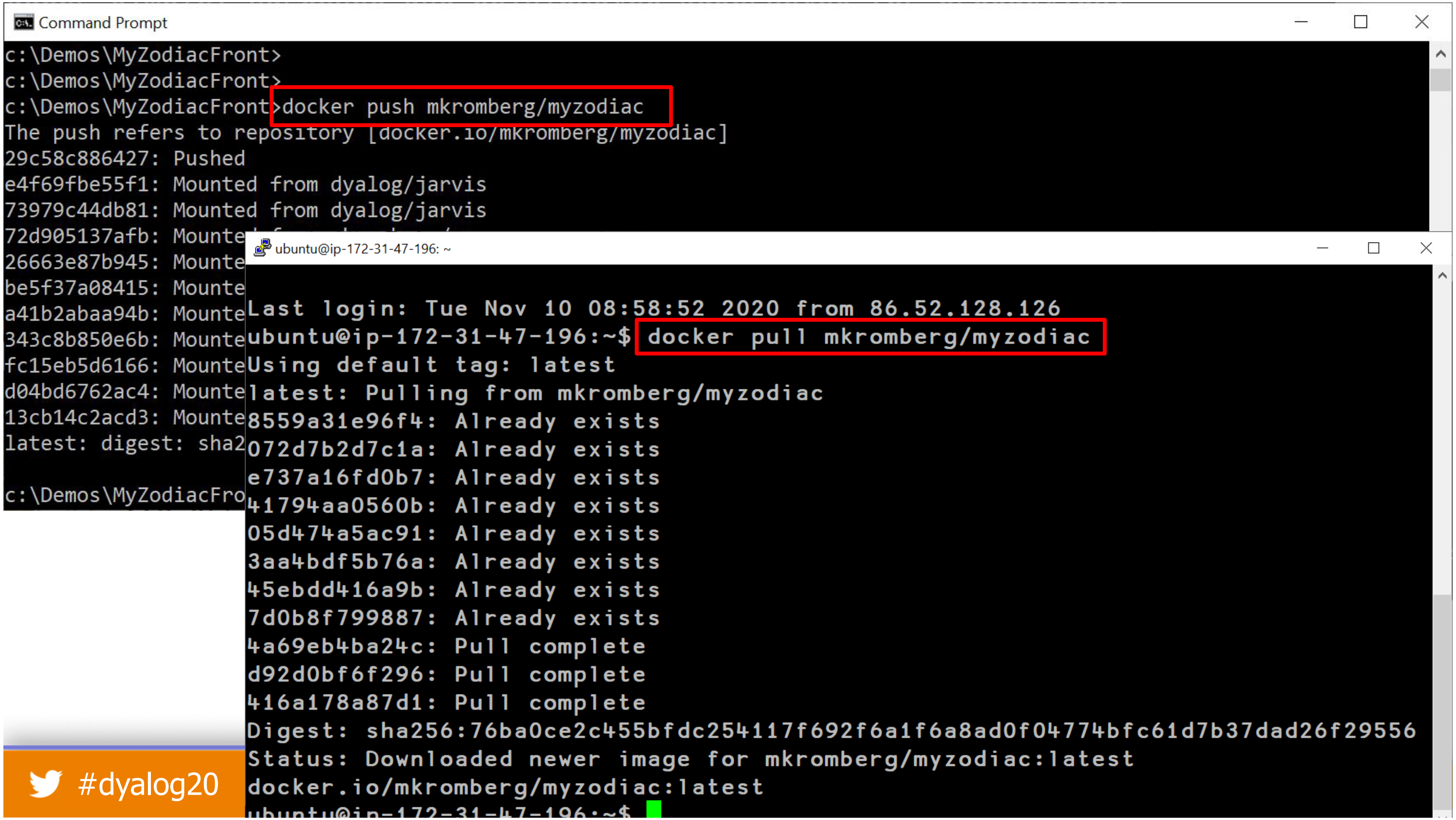
Method to Execute: GetSign

JSON Data:

```
[3,31]
```

Response

```
"Ram"
```



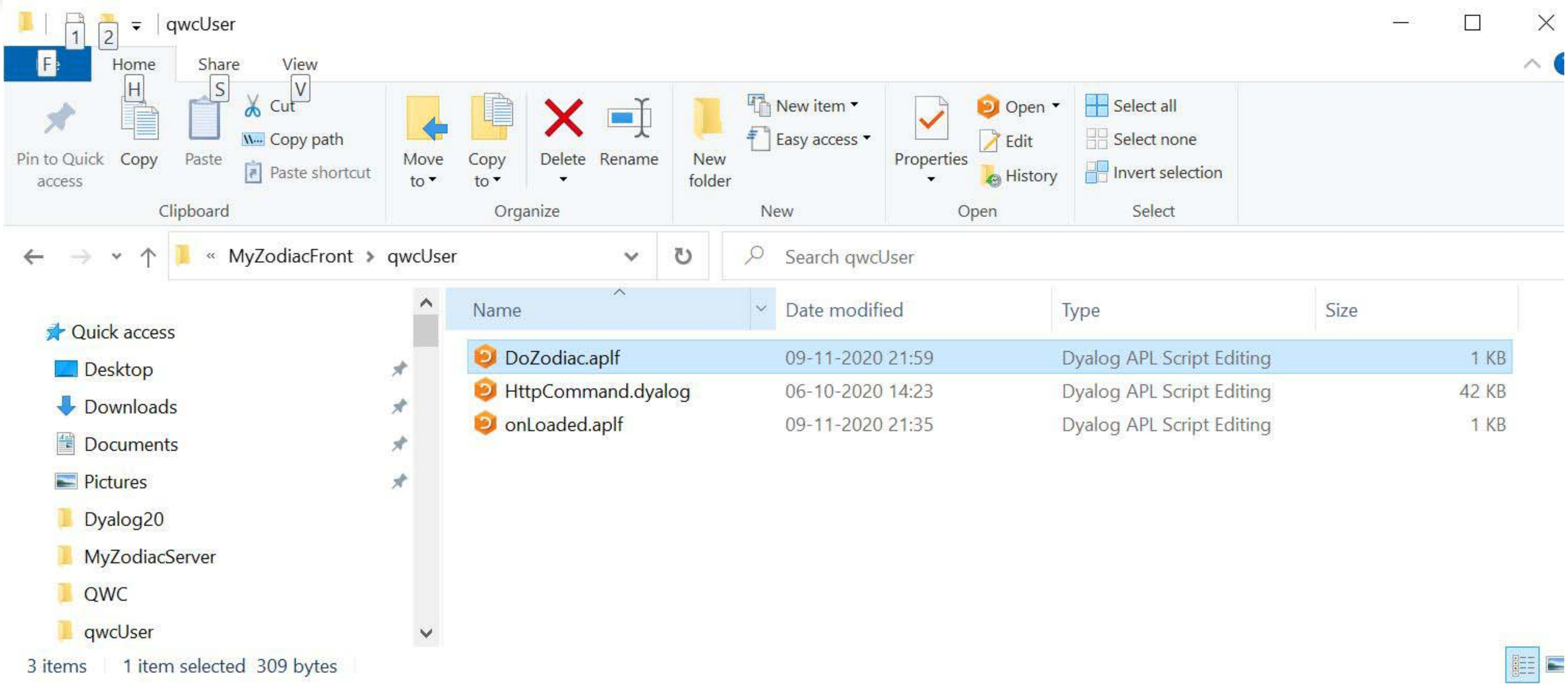
```
c:\Demos\MyZodiacFront>
c:\Demos\MyZodiacFront>
c:\Demos\MyZodiacFront>docker push mkromberg/myzodiac
The push refers to repository [docker.io/mkromberg/myzodiac]
29c58c886427: Pushed
e4f69fbe55f1: Mounted from dyalog/jarvis
73979c44db81: Mounted from dyalog/jarvis
72d905137afb: Mounte
26663e87b945: Mounte
be5f37a08415: Mounte
a41b2abaa94b: Mounte
343c8b850e6b: Mounte
fc15eb5d6166: Mounte
d04bd6762ac4: Mounte
13cb14c2acd3: Mounte
latest: digest: sha2

c:\Demos\MyZodiacFro
```

```
Last login: Tue Nov 10 08:58:52 2020 from 86.52.128.126
ubuntu@ip-172-31-47-196:~$ docker pull mkromberg/myzodiac
Using default tag: latest
latest: Pulling from mkromberg/myzodiac
8559a31e96f4: Already exists
072d7b2d7c1a: Already exists
e737a16fd0b7: Already exists
41794aa0560b: Already exists
05d474a5ac91: Already exists
3aa4bdf5b76a: Already exists
45ebdd416a9b: Already exists
7d0b8f799887: Already exists
4a69eb4ba24c: Pull complete
d92d0bf6f296: Pull complete
416a178a87d1: Pull complete
Digest: sha256:76ba0ce2c455bfdc254117f692f6a1f6a8ad0f04774bfc61d7b37dad26f29556
Status: Downloaded newer image for mkromberg/myzodiac:latest
docker.io/mkromberg/myzodiac:latest
ubuntu@ip-172-31-47-196:~$
```

#dyalog20

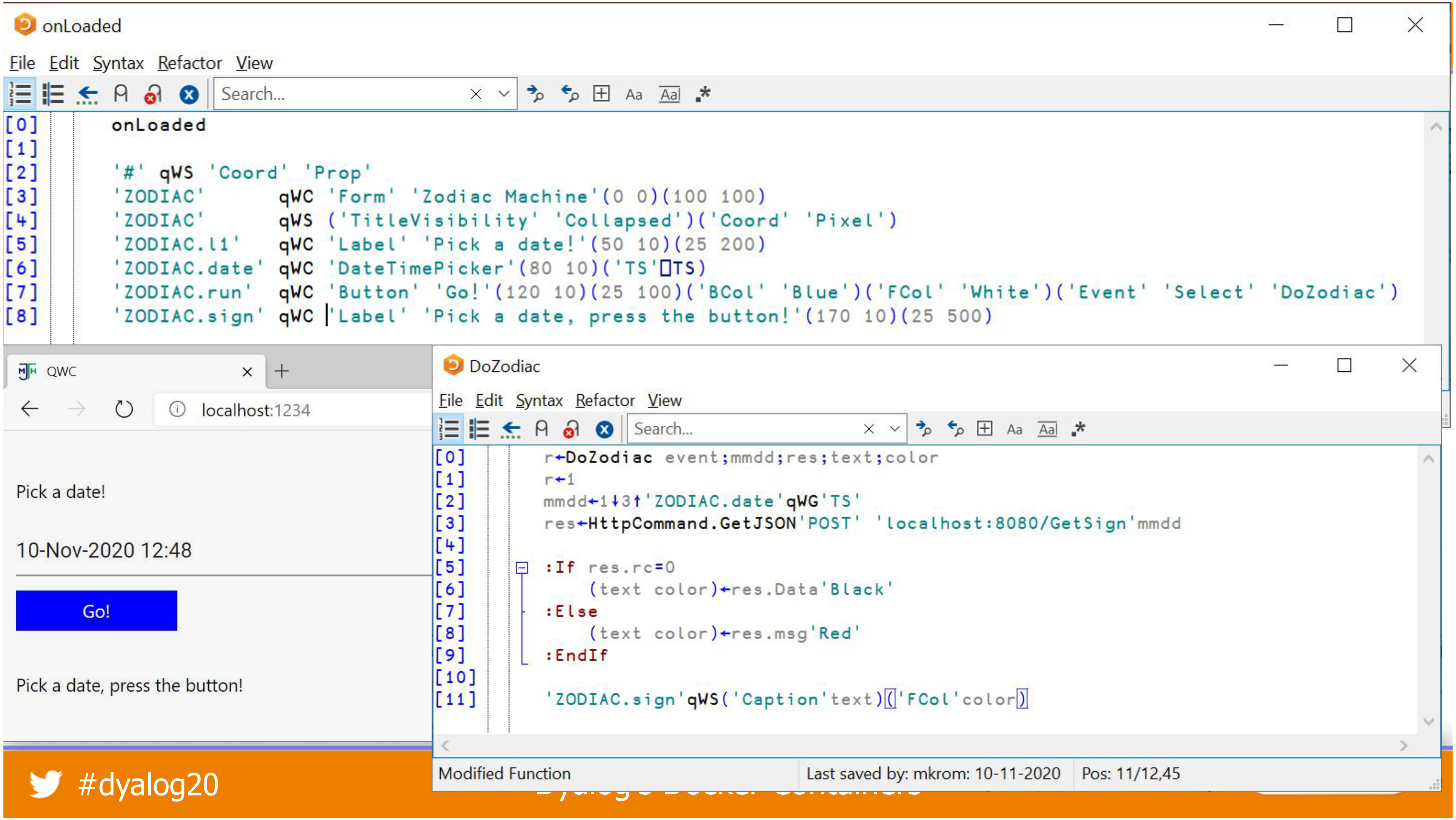
Collaborating Containers

- We will use `qWC` to put a frontend on the `myzodiac` server



Name	Date modified	Type	Size
DoZodiac.aplf	09-11-2020 21:59	Dyalog APL Script Editing	1 KB
HttpCommand.dyalog	06-10-2020 14:23	Dyalog APL Script Editing	42 KB
onLoaded.aplf	09-11-2020 21:35	Dyalog APL Script Editing	1 KB

3 items 1 item selected 309 bytes



```
onLoaded

'#' qWS 'Coord' 'Prop'
'ZODIAC'      qWC 'Form' 'Zodiac Machine'(0 0)(100 100)
'ZODIAC'      qWS ('TitleVisibility' 'Collapsed')('Coord' 'Pixel')
'ZODIAC.l1'   qWC 'Label' 'Pick a date!'(50 10)(25 200)
'ZODIAC.date' qWC 'DateTimePicker'(80 10)('TS'⎕TS)
'ZODIAC.run'  qWC 'Button' 'Go!'(120 10)(25 100)('BCol' 'Blue')('FCol' 'White')('Event' 'Select' 'DoZodiac')
'ZODIAC.sign' qWC 'Label' 'Pick a date, press the button!'(170 10)(25 500)
```

Pick a date!

10-Nov-2020 12:48

Pick a date, press the button!

```
r←DoZodiac event;mmdd;res;text;color
r←1
mmdd←1↓3↑'ZODIAC.date'qWG'TS'
res←HttpCommand.GetJSON'POST' 'localhost:8080/GetSign'mmdd

:If res.rc=0
    (text color)←res.Data'Black'
:Else
    (text color)←res.msg'Red'
:EndIf

'ZODIAC.sign'qWS('Caption'text)('FCol'color)
```

Modified Function | Last saved by: mkrom: 10-11-2020 | Pos: 11/12,45

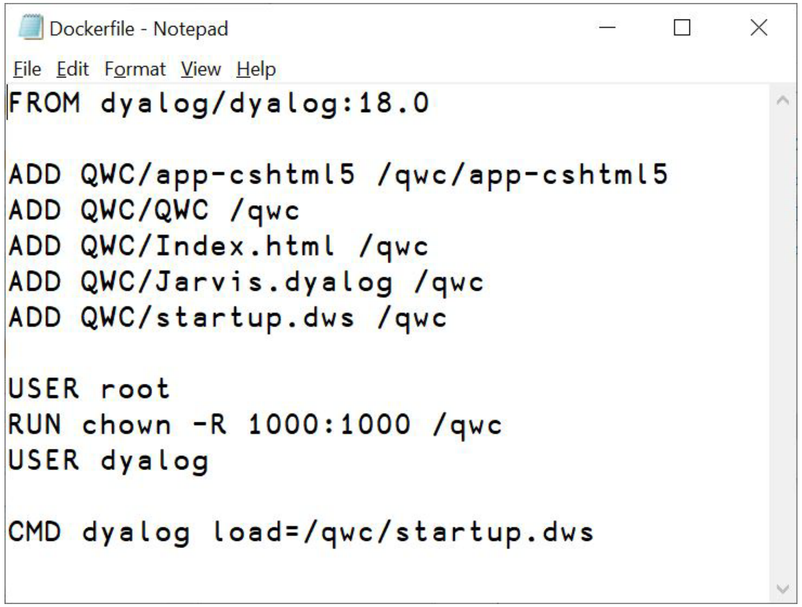
Let's build a qWC Container

Dyalog APL plus some QWC folders:

```
docker build . -t mkromberg/qwc
```

Maybe someday this will be

```
mjhsoftware/qwc
```



```
FROM dyalog/dyalog:18.0

ADD QWC/app-cshtml5 /qwc/app-cshtml5
ADD QWC/QWC /qwc
ADD QWC/Index.html /qwc
ADD QWC/Jarvis.dyalog /qwc
ADD QWC/startup.dws /qwc

USER root
RUN chown -R 1000:1000 /qwc
USER dyalog

CMD dyalog load=/qwc/startup.dws
```

To start the application

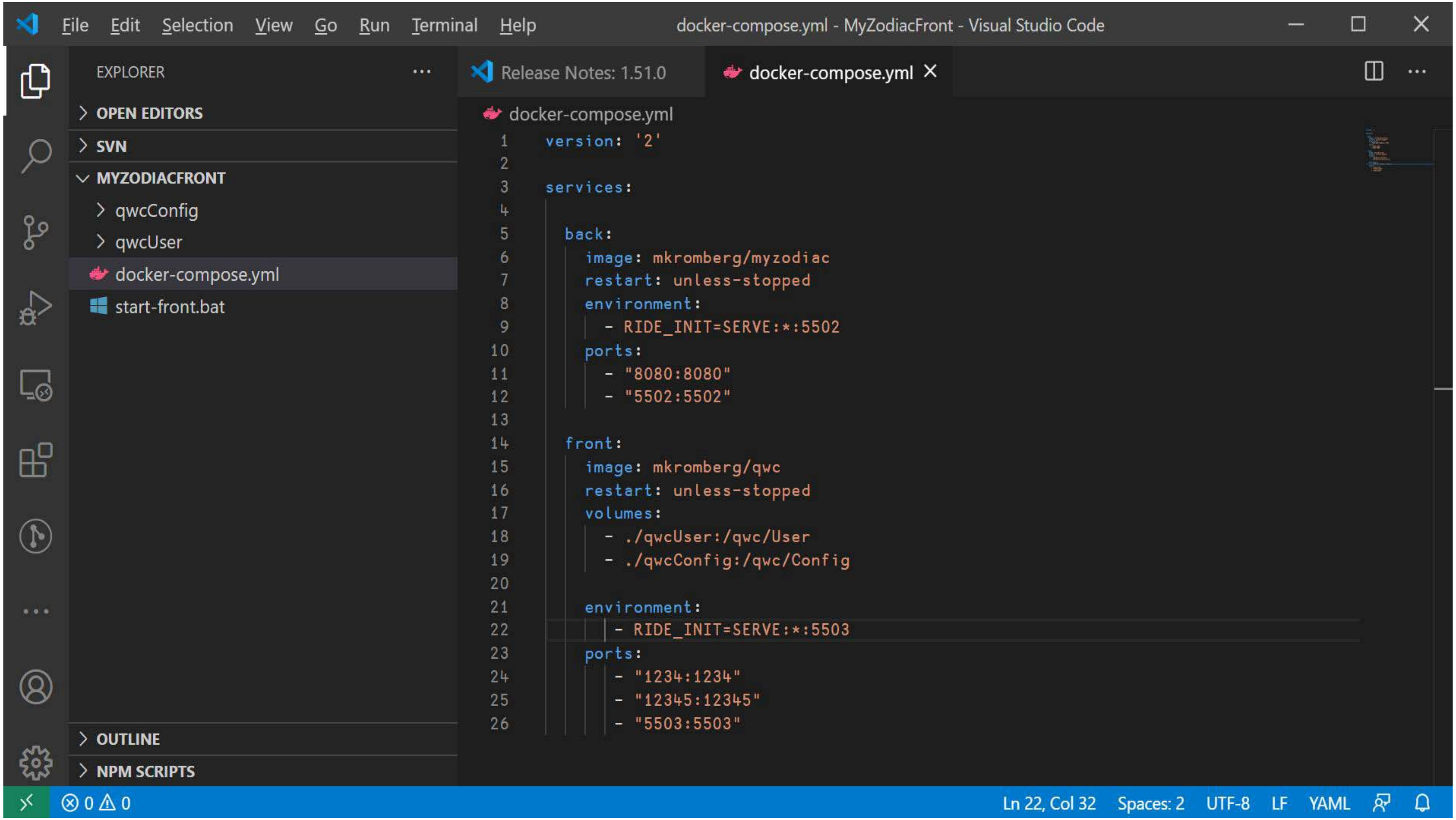
- Start Jarvis "backend"

  ```
  docker run -it -p 8080:8080 mkromberg/myzodiac
  ```

- Start QWC frontend

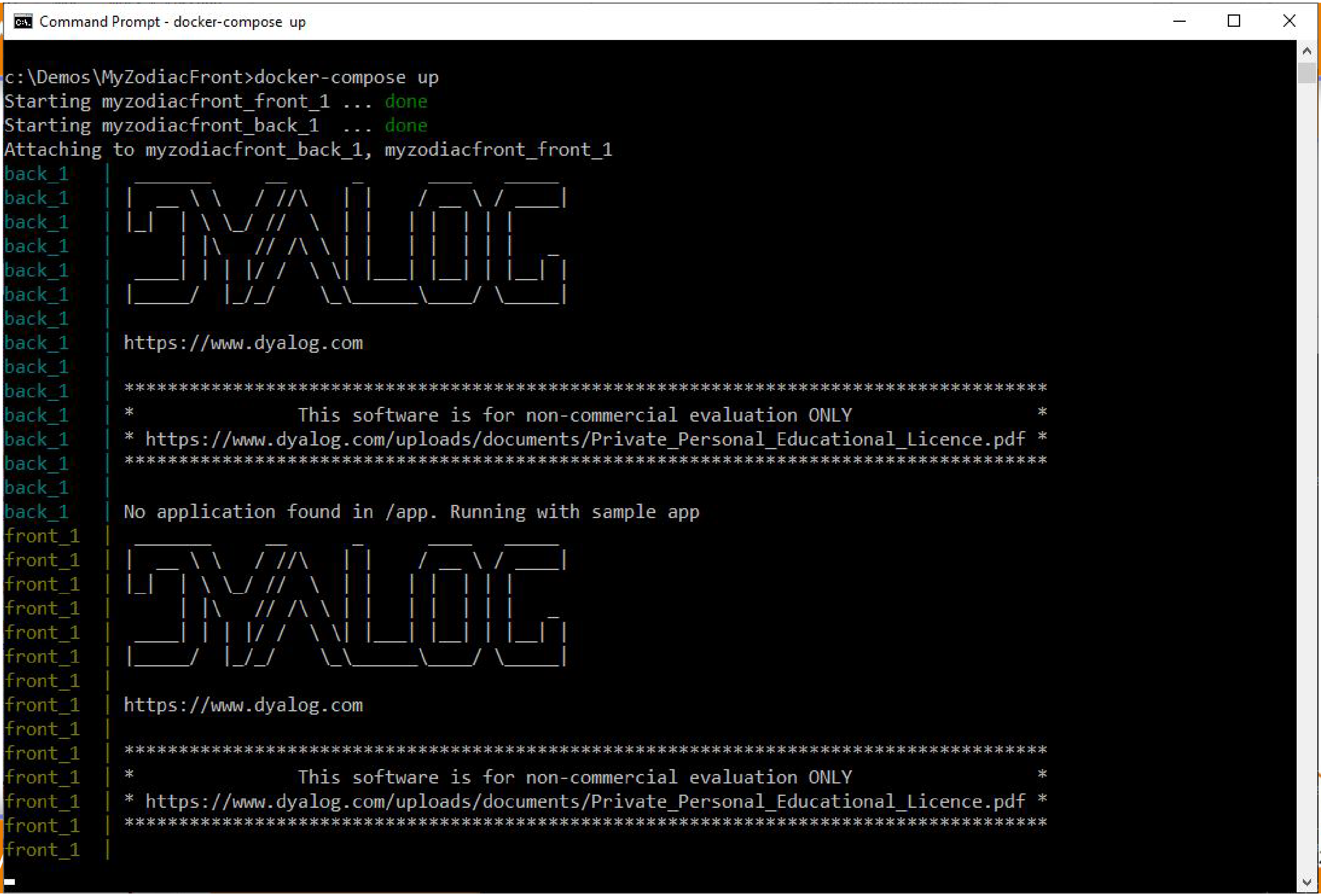
  ```
  docker run -it
  -v qwcuser:/qwc/User
  -v qwcconfig:/qwc/Config
  -p 1234:1234 -p 12345:12345
  mkromberg/qwc
  ```

... or use "docker-compose" ...



```yaml
version: '2'

services:

  back:
    image: mkromberg/myzodiac
    restart: unless-stopped
    environment:
      - RIDE_INIT=SERVE:*:5502
    ports:
      - "8080:8080"
      - "5502:5502"

  front:
    image: mkromberg/qwc
    restart: unless-stopped
    volumes:
      - ./qwcUser:/qwc/User
      - ./qwcConfig:/qwc/Config

    environment:
       - RIDE_INIT=SERVE:*:5503
    ports:
       - "1234:1234"
       - "12345:12345"
       - "5503:5503"
```

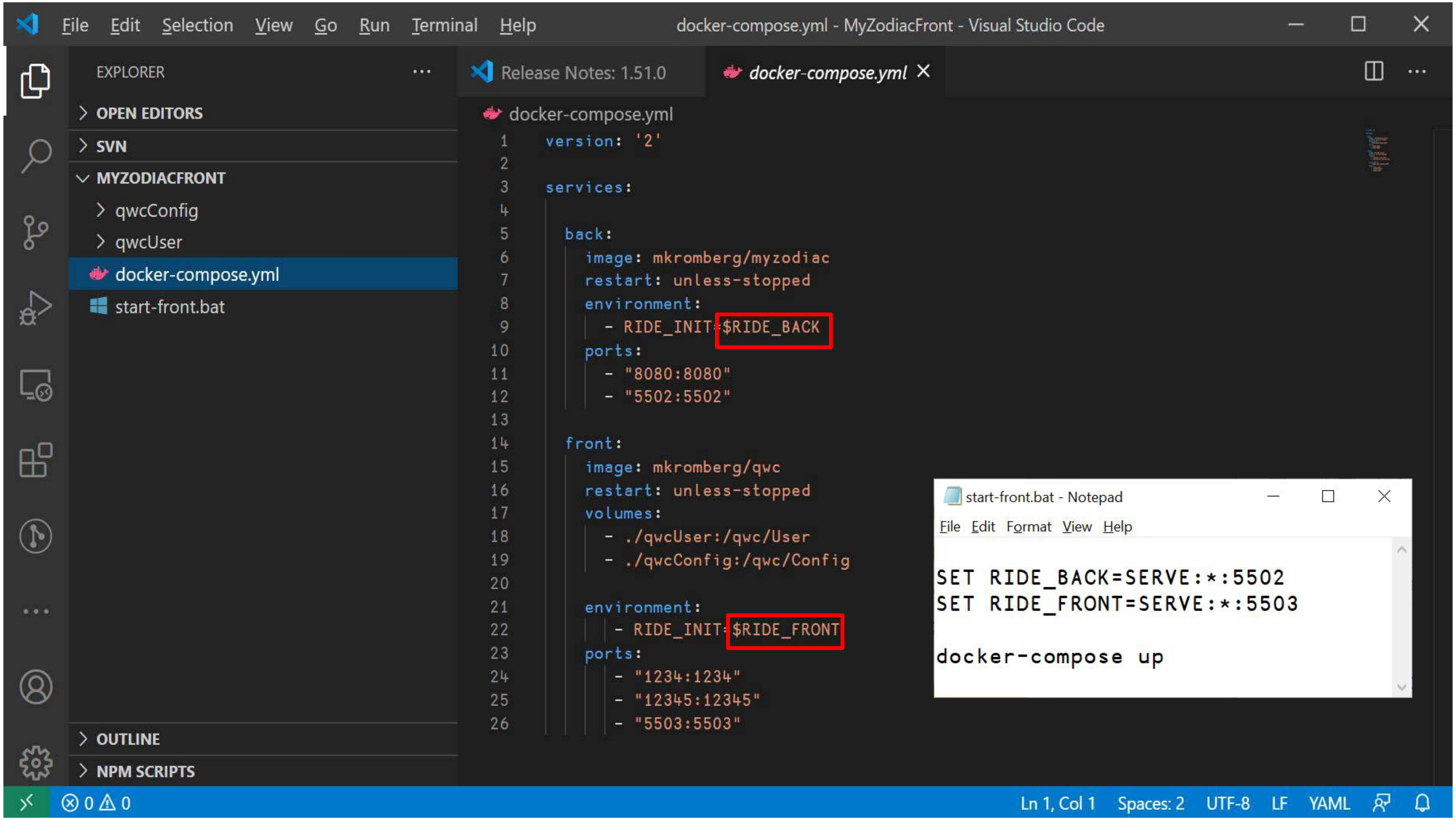
```
docker-compose up
```



Command Prompt - docker-compose up

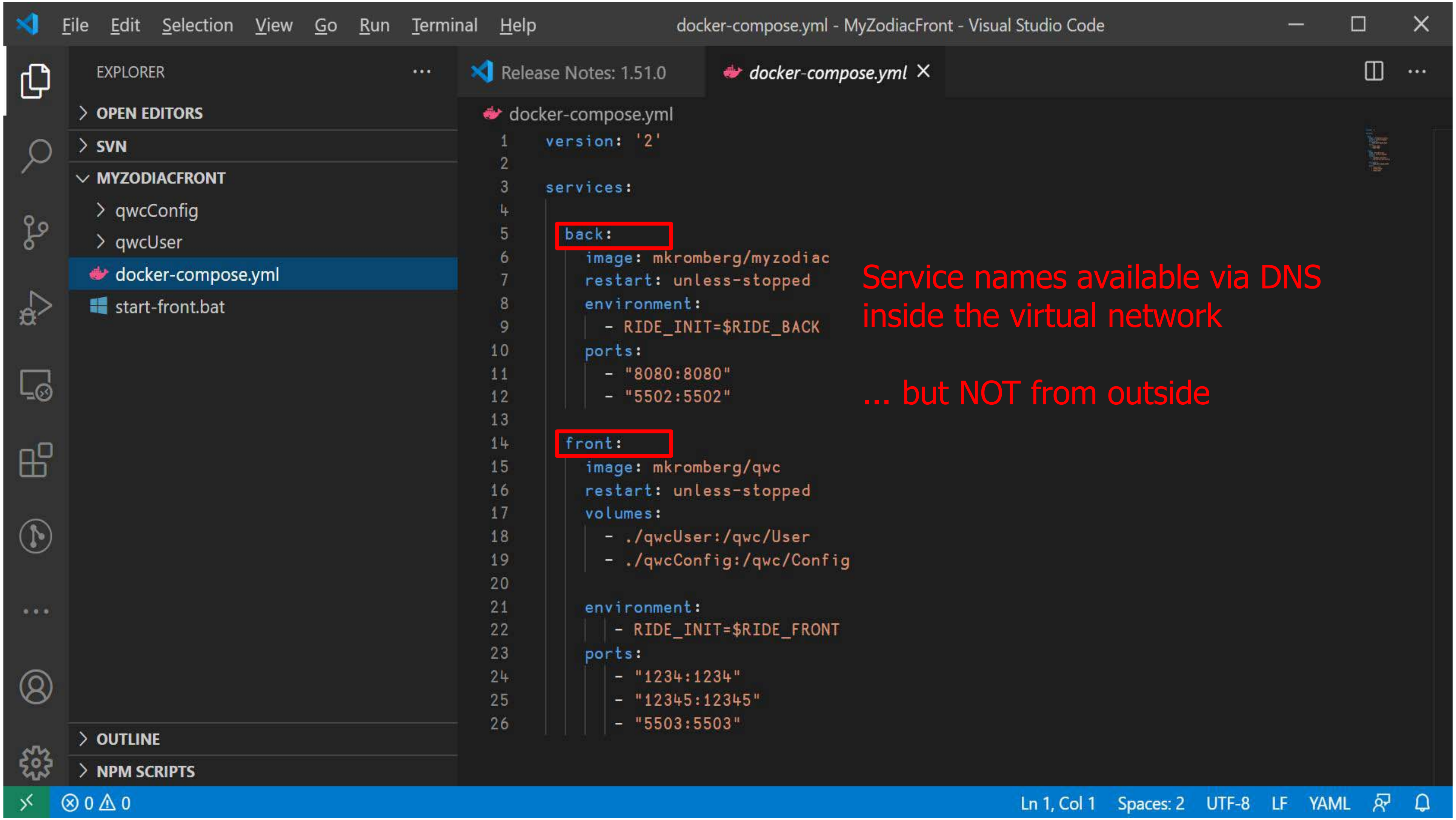
```
c:\Demos\MyZodiacFront>docker-compose up
Starting myzodiacfront_front_1 ... done
Starting myzodiacfront_back_1  ... done
Attaching to myzodiacfront_back_1, myzodiacfront_front_1
back_1   |  _______     __      _      ____   _____
back_1   | |  __ \ \   / //\   | |    / __ \ / ____|
back_1   | |_|  | \ \_/ //  \  | |   | |  | | |
back_1   |      | |\   // /\ \ | |   | |  | | |   _
back_1   |  ____| | | |/ / /\ \| |___| |__| | |__| |
back_1   | |_____/  |_/_/    \_\______\____/ \_____|
back_1   |
back_1   | https://www.dyalog.com
back_1   |
back_1   | *************************************************************************************
back_1   | *               This software is for non-commercial evaluation ONLY                 *
back_1   | * https://www.dyalog.com/uploads/documents/Private_Personal_Educational_Licence.pdf *
back_1   | *************************************************************************************
back_1   |
back_1   | No application found in /app. Running with sample app
front_1  |  _______     __      _      ____   _____
front_1  | |  __ \ \   / //\   | |    / __ \ / ____|
front_1  | |_|  | \ \_/ //  \  | |   | |  | | |
front_1  |      | |\   // /\ \ | |   | |  | | |   _
front_1  |  ____| | | |/ / /\ \| |___| |__| | |__| |
front_1  | |_____/  |_/_/    \_\______\____/ \_____|
front_1  |
front_1  | https://www.dyalog.com
front_1  |
front_1  | *************************************************************************************
front_1  | *               This software is for non-commercial evaluation ONLY                 *
front_1  | * https://www.dyalog.com/uploads/documents/Private_Personal_Educational_Licence.pdf *
front_1  | *************************************************************************************
front_1  |
```

Demo 4 docker-compose up



```yaml
version: '2'

services:

  back:
    image: mkromberg/myzodiac
    restart: unless-stopped
    environment:
      - RIDE_INIT=$RIDE_BACK
    ports:
      - "8080:8080"
      - "5502:5502"

  front:
    image: mkromberg/qwc
    restart: unless-stopped
    volumes:
      - ./qwcUser:/qwc/User
      - ./qwcConfig:/qwc/Config

    environment:
       - RIDE_INIT=$RIDE_FRONT
    ports:
       - "1234:1234"
       - "12345:12345"
       - "5503:5503"
```

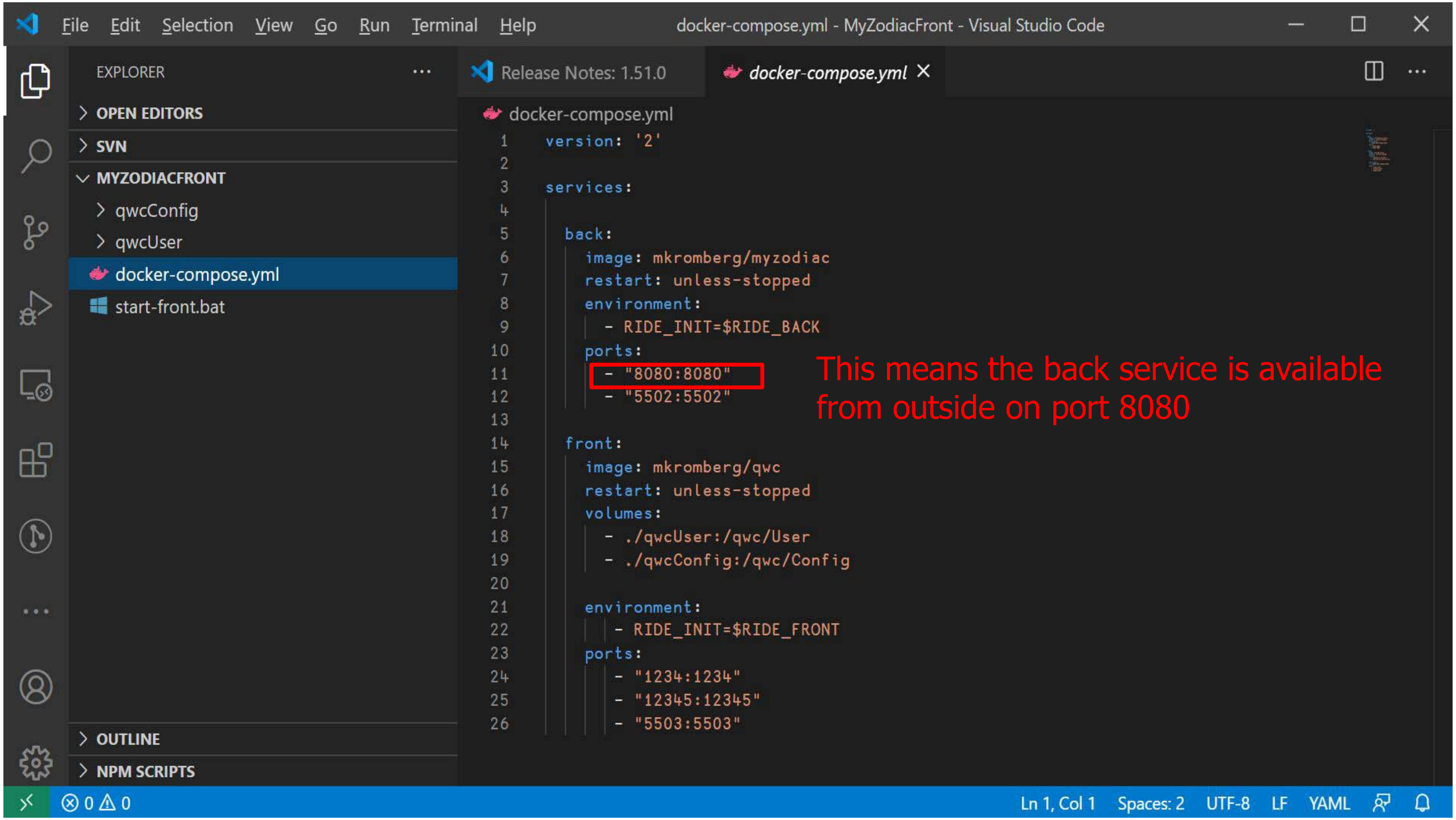
```
SET RIDE_BACK=SERVE:*:5502
SET RIDE_FRONT=SERVE:*:5503

docker-compose up
```



```
version: '2'

services:

  back:
    image: mkromberg/myzodiac
    restart: unless-stopped
    environment:
      - RIDE_INIT=$RIDE_BACK
    ports:
      - "8080:8080"
      - "5502:5502"

  front:
    image: mkromberg/qwc
    restart: unless-stopped
    volumes:
      - ./qwcUser:/qwc/User
      - ./qwcConfig:/qwc/Config

    environment:
      - RIDE_INIT=$RIDE_FRONT
    ports:
      - "1234:1234"
      - "12345:12345"
      - "5503:5503"
```

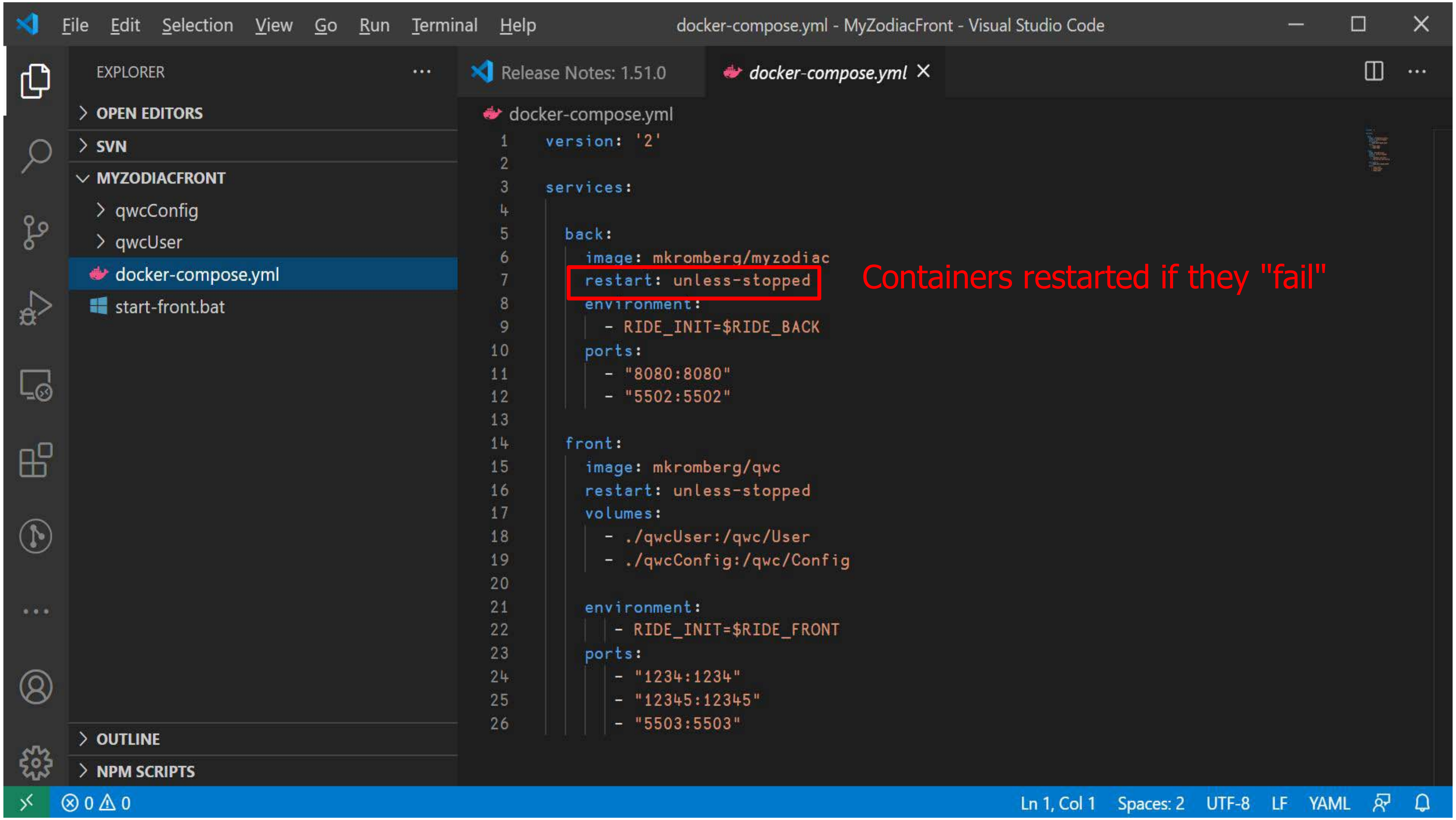
Service names available via DNS inside the virtual network

... but NOT from outside



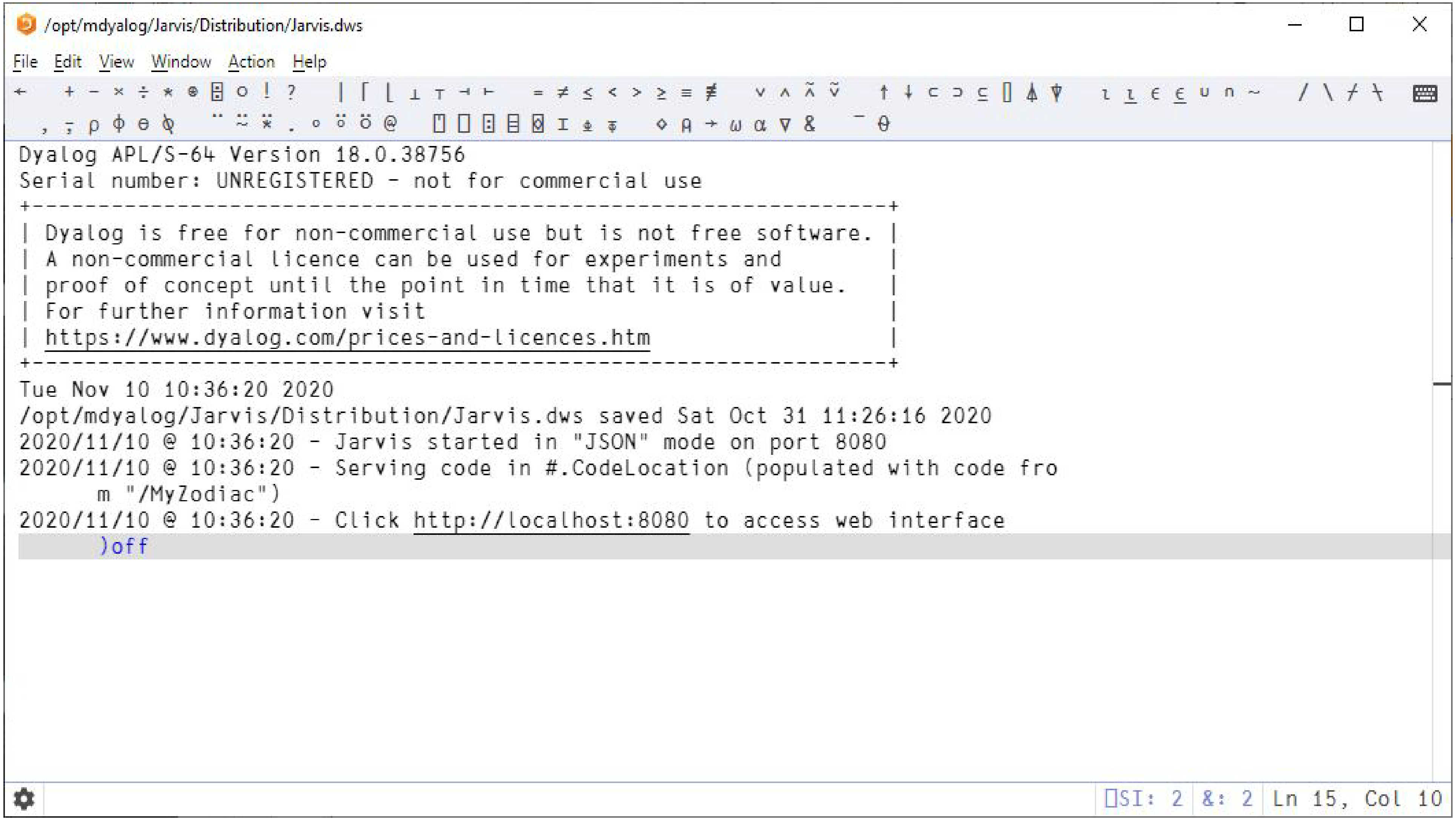
```yaml
version: '2'

services:

  back:
    image: mkromberg/myzodiac
    restart: unless-stopped
    environment:
      - RIDE_INIT=$RIDE_BACK
    ports:
      - "8080:8080"
      - "5502:5502"

  front:
    image: mkromberg/qwc
    restart: unless-stopped
    volumes:
      - ./qwcUser:/qwc/User
      - ./qwcConfig:/qwc/Config

    environment:
       - RIDE_INIT=$RIDE_FRONT
    ports:
       - "1234:1234"
       - "12345:12345"
       - "5503:5503"
```

This means the back service is available from outside on port 8080

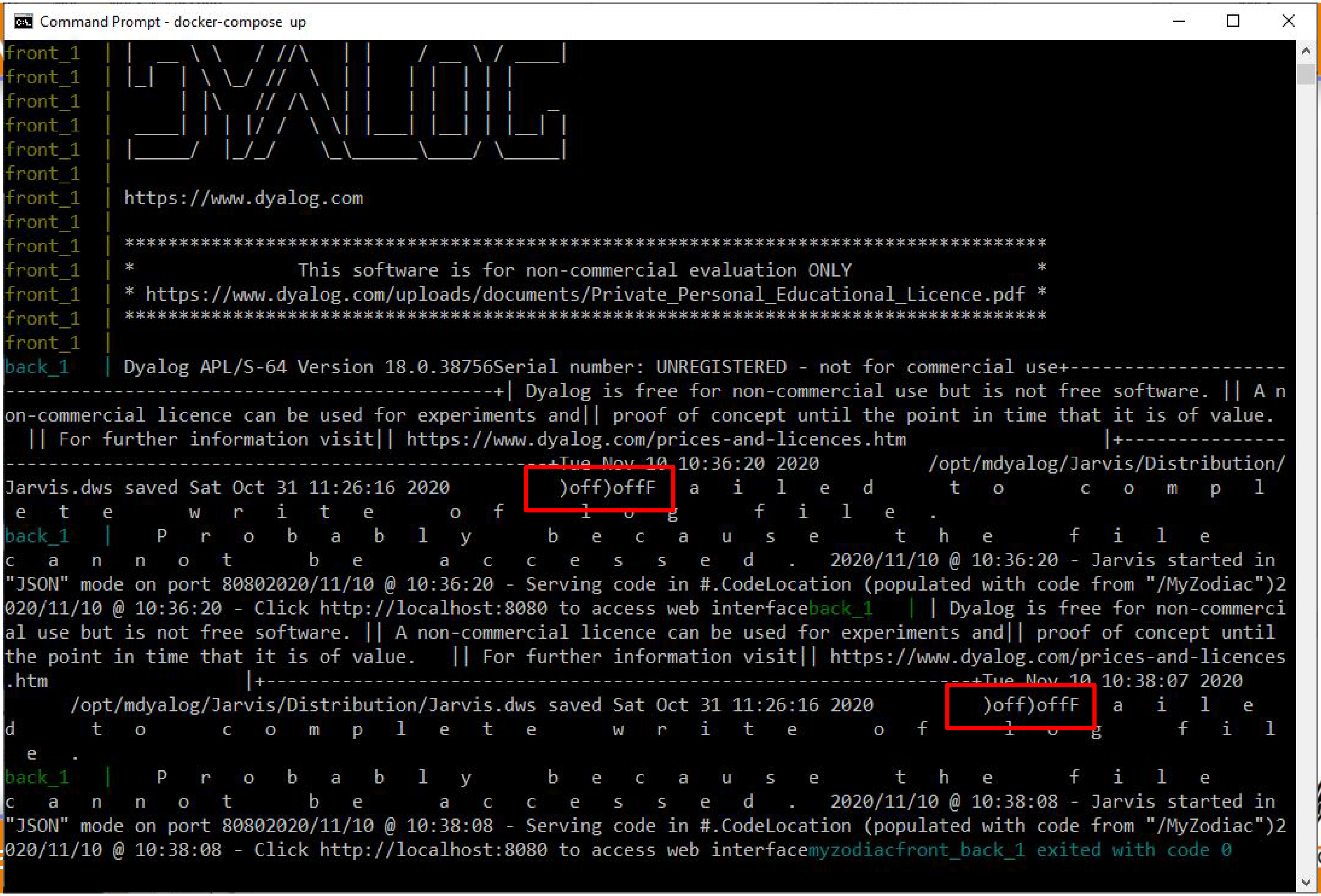


```yaml
version: '2'

services:

  back:
    image: mkromberg/myzodiac
    restart: unless-stopped
    environment:
      - RIDE_INIT=$RIDE_BACK
    ports:
      - "8080:8080"
      - "5502:5502"

  front:
    image: mkromberg/qwc
    restart: unless-stopped
    volumes:
      - ./qwcUser:/qwc/User
      - ./qwcConfig:/qwc/Config

    environment:
      - RIDE_INIT=$RIDE_FRONT
    ports:
      - "1234:1234"
      - "12345:12345"
      - "5503:5503"
```

Containers restarted if they "fail"


/opt/mdyalog/Jarvis/Distribution/Jarvis.dws
File Edit View Window Action Help
⎕SI: 2 &: 2 Ln 15, Col 10

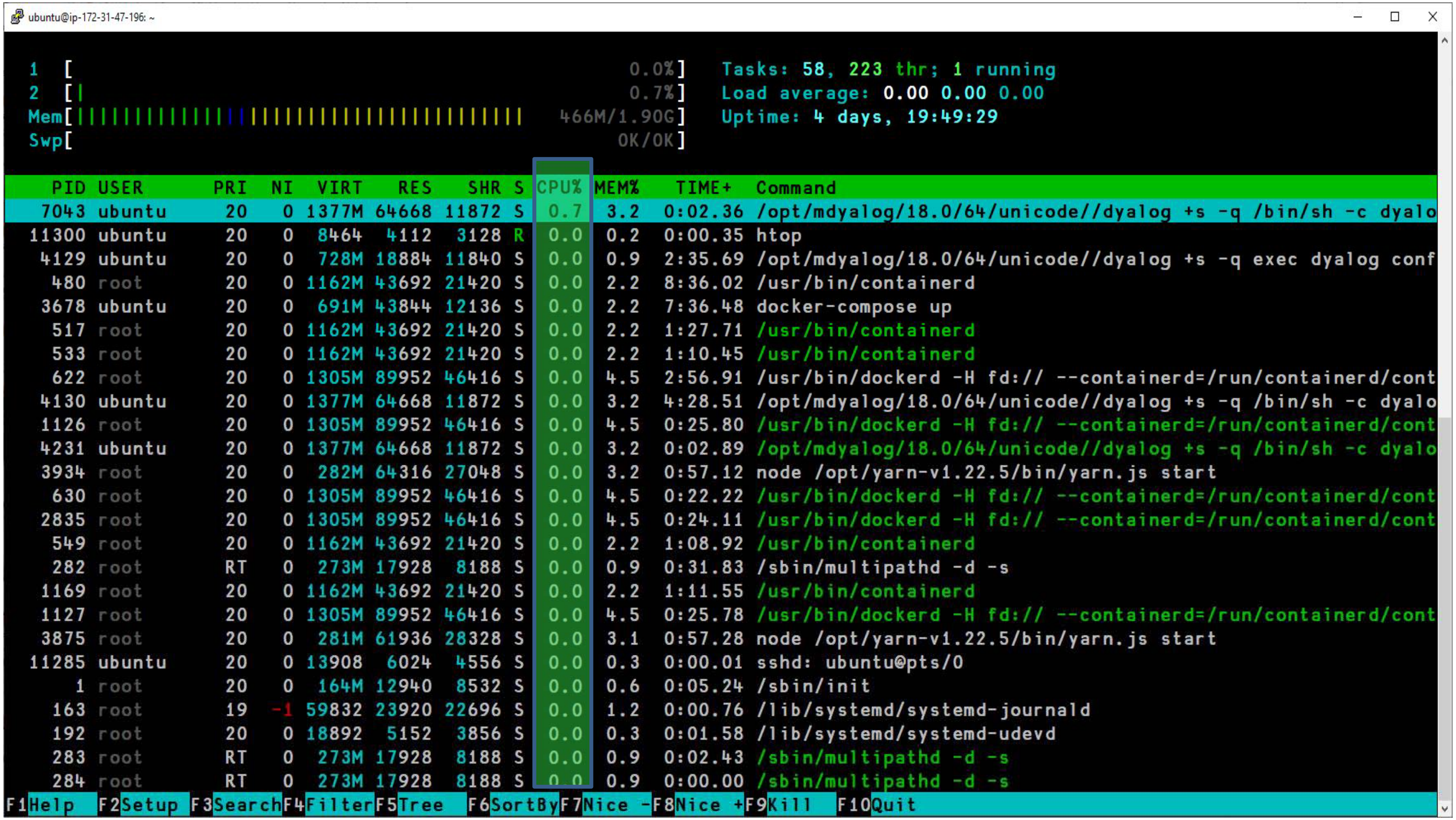
```
Dyalog APL/S-64 Version 18.0.38756
Serial number: UNREGISTERED - not for commercial use
+-----------------------------------------------------------------+
| Dyalog is free for non-commercial use but is not free software. |
| A non-commercial licence can be used for experiments and        |
| proof of concept until the point in time that it is of value.   |
| For further information visit                                   |
| https://www.dyalog.com/prices-and-licences.htm                  |
+-----------------------------------------------------------------+
Tue Nov 10 10:36:20 2020
/opt/mdyalog/Jarvis/Distribution/Jarvis.dws saved Sat Oct 31 11:26:16 2020
2020/11/10 @ 10:36:20 - Jarvis started in "JSON" mode on port 8080
2020/11/10 @ 10:36:20 - Serving code in #.CodeLocation (populated with code fro
      m "/MyZodiac")
2020/11/10 @ 10:36:20 - Click http://localhost:8080 to access web interface
      )off
```



```
front_1  | |  __ \ \   / //\   | |     /  __ \ /  ____|
front_1  | |_|  | \ \_/ //  \  | |    | |  | | |
front_1  |      | |\   // /\ \ | |    | |  | | |  _
front_1  |  ____| | | |/ /  \ \| |____| |__| | |__| |
front_1  | |_____/  |_/_/    \_\______\____/ \_____|
front_1  |
front_1  | https://www.dyalog.com
front_1  |
front_1  | ********************************************************************************
front_1  | *                This software is for non-commercial evaluation ONLY              *
front_1  | * https://www.dyalog.com/uploads/documents/Private_Personal_Educational_Licence.pdf *
front_1  | ********************************************************************************
front_1  |
back_1   | Dyalog APL/S-64 Version 18.0.38756Serial number: UNREGISTERED - not for commercial use+--------------------
---------------------------------------------+| Dyalog is free for non-commercial use but is not free software. || A n
on-commercial licence can be used for experiments and|| proof of concept until the point in time that it is of value.
  || For further information visit|| https://www.dyalog.com/prices-and-licences.htm                    |+--------------
-------------------------------------------------+Tue Nov 10 10:36:20 2020         /opt/mdyalog/Jarvis/Distribution/
Jarvis.dws saved Sat Oct 31 11:26:16 2020         )off)offF  a   i   l   e   d     t   o     c   o   m   p   l
 e   t   e     w   r   i   t   e     o   f     l   o   g     f   i   l   e   .
back_1   |    P   r   o   b   a   b   l   y     b   e   c   a   u   s   e     t   h   e     f   i   l   e
c   a   n   n   o   t     b   e     a   c   c   e   s   s   e   d   .   2020/11/10 @ 10:36:20 - Jarvis started in
"JSON" mode on port 80802020/11/10 @ 10:36:20 - Serving code in #.CodeLocation (populated with code from "/MyZodiac")2
020/11/10 @ 10:36:20 - Click http://localhost:8080 to access web interfaceback_1   | | Dyalog is free for non-commerci
al use but is not free software. || A non-commercial licence can be used for experiments and|| proof of concept until
the point in time that it is of value.   || For further information visit|| https://www.dyalog.com/prices-and-licences
.htm                  |+--------------------------------------------------------------------+Tue Nov 10 10:38:07 2020
      /opt/mdyalog/Jarvis/Distribution/Jarvis.dws saved Sat Oct 31 11:26:16 2020          )off)offF  a   i   l   e
d     t   o     c   o   m   p   l   e   t   e     w   r   i   t   e     o   f     l   o   g     f   i   l
  e   .
back_1   |    P   r   o   b   a   b   l   y     b   e   c   a   u   s   e     t   h   e     f   i   l   e
c   a   n   n   o   t     b   e     a   c   c   e   s   s   e   d   .   2020/11/10 @ 10:38:08 - Jarvis started in
"JSON" mode on port 80802020/11/10 @ 10:38:08 - Serving code in #.CodeLocation (populated with code from "/MyZodiac")2
020/11/10 @ 10:38:08 - Click http://localhost:8080 to access web interfacemyzodiacfront_back_1 exited with code 0
```

docker-compose can also do

- Load balancing by starting multiple copies of the same image and multiplexing between them
- Help with continuous integration by pulling new versions of the images
- and much more that I have not learned about yet

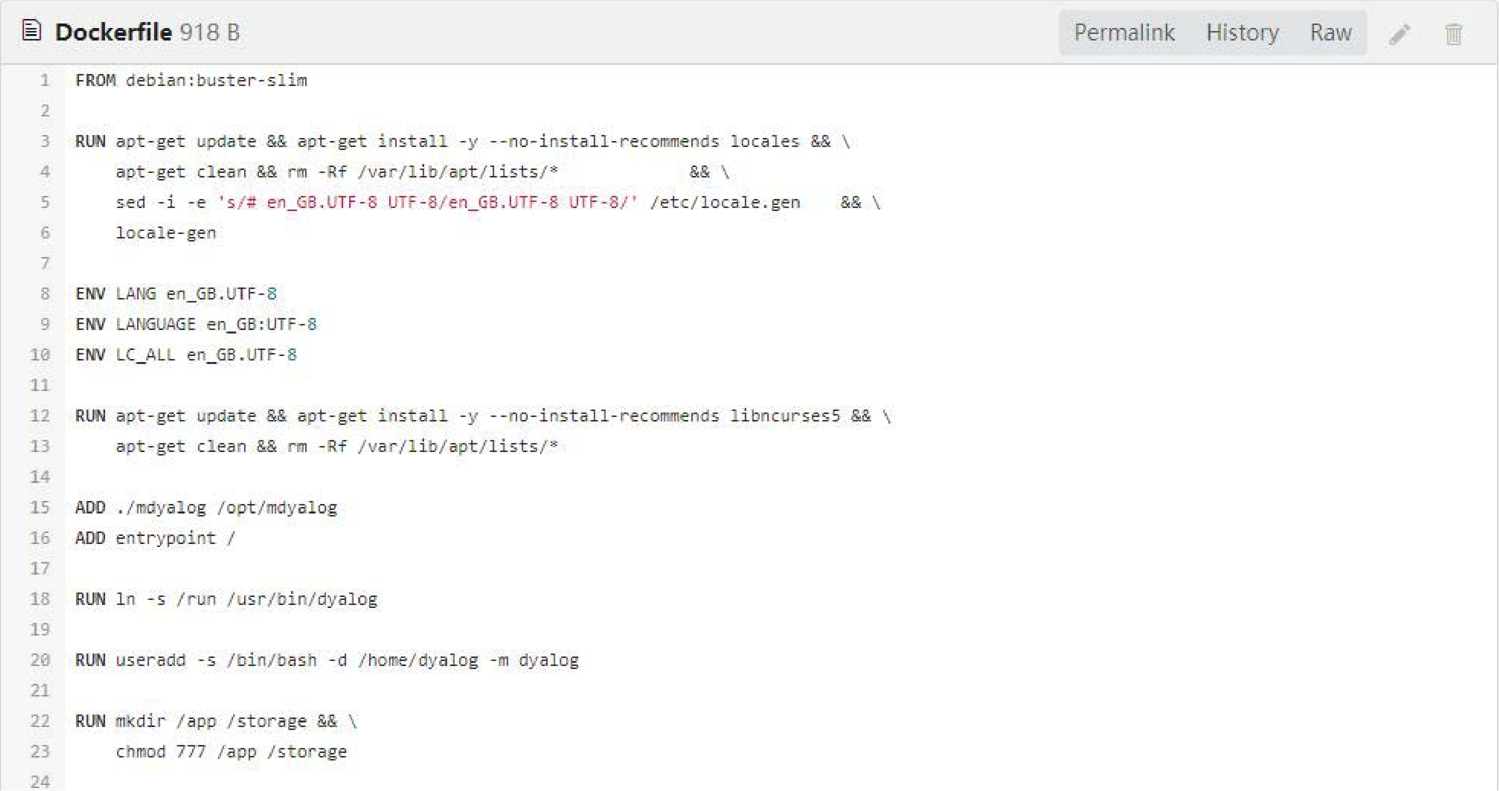


```
  1  [                                       0.0%]   Tasks: 58, 223 thr; 1 running
  2  [|                                      0.7%]   Load average: 0.00 0.00 0.00
  Mem[|||||||||||||||||||||||||||||||||||||||   466M/1.90G]   Uptime: 4 days, 19:49:29
  Swp[                                      0K/0K]

    PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
   7043 ubuntu     20   0 1377M 64668 11872 S  0.7  3.2  0:02.36 /opt/mdyalog/18.0/64/unicode//dyalog +s -q /bin/sh -c dyalo
  11300 ubuntu     20   0  8464  4112  3128 R  0.0  0.2  0:00.35 htop
   4129 ubuntu     20   0  728M 18884 11840 S  0.0  0.9  2:35.69 /opt/mdyalog/18.0/64/unicode//dyalog +s -q exec dyalog conf
    480 root       20   0 1162M 43692 21420 S  0.0  2.2  8:36.02 /usr/bin/containerd
   3678 ubuntu     20   0  691M 43844 12136 S  0.0  2.2  7:36.48 docker-compose up
    517 root       20   0 1162M 43692 21420 S  0.0  2.2  1:27.71 /usr/bin/containerd
    533 root       20   0 1162M 43692 21420 S  0.0  2.2  1:10.45 /usr/bin/containerd
    622 root       20   0 1305M 89952 46416 S  0.0  4.5  2:56.91 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont
   4130 ubuntu     20   0 1377M 64668 11872 S  0.0  3.2  4:28.51 /opt/mdyalog/18.0/64/unicode//dyalog +s -q /bin/sh -c dyalo
   1126 root       20   0 1305M 89952 46416 S  0.0  4.5  0:25.80 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont
   4231 ubuntu     20   0 1377M 64668 11872 S  0.0  3.2  0:02.89 /opt/mdyalog/18.0/64/unicode//dyalog +s -q /bin/sh -c dyalo
   3934 root       20   0  282M 64316 27048 S  0.0  3.2  0:57.12 node /opt/yarn-v1.22.5/bin/yarn.js start
    630 root       20   0 1305M 89952 46416 S  0.0  4.5  0:22.22 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont
   2835 root       20   0 1305M 89952 46416 S  0.0  4.5  0:24.11 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont
    549 root       20   0 1162M 43692 21420 S  0.0  2.2  1:08.92 /usr/bin/containerd
    282 root       RT   0  273M 17928  8188 S  0.0  0.9  0:31.83 /sbin/multipathd -d -s
   1169 root       20   0 1162M 43692 21420 S  0.0  2.2  1:11.55 /usr/bin/containerd
   1127 root       20   0 1305M 89952 46416 S  0.0  4.5  0:25.78 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont
   3875 root       20   0  281M 61936 28328 S  0.0  3.1  0:57.28 node /opt/yarn-v1.22.5/bin/yarn.js start
  11285 ubuntu     20   0 13908  6024  4556 S  0.0  0.3  0:00.01 sshd: ubuntu@pts/0
      1 root       20   0  164M 12940  8532 S  0.0  0.6  0:05.24 /sbin/init
    163 root       19  -1 59832 23920 22696 S  0.0  1.2  0:00.76 /lib/systemd/systemd-journald
    192 root       20   0 18892  5152  3856 S  0.0  0.3  0:01.58 /lib/systemd/systemd-udevd
    283 root       RT   0  273M 17928  8188 S  0.0  0.9  0:02.43 /sbin/multipathd -d -s
    284 root       RT   0  273M 17928  8188 S  0.0  0.9  0:00.00 /sbin/multipathd -d -s
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Relevant v19.0 Goals

- Completely rework I/O handling to allow
  - Support RIDE without "-it" switch
- RIDE improvements in multi-threaded server environments
- Documentation for all the public dyalog containers
- Open-source demo apps showing how to build secure, load balanced apps in APL

We will publish our Dockerfiles on GitHub



Dockerfile 918 B

```
FROM debian:buster-slim

RUN apt-get update && apt-get install -y --no-install-recommends locales && \
    apt-get clean && rm -Rf /var/lib/apt/lists/*             && \
    sed -i -e 's/# en_GB.UTF-8 UTF-8/en_GB.UTF-8 UTF-8/' /etc/locale.gen    && \
    locale-gen

ENV LANG en_GB.UTF-8
ENV LANGUAGE en_GB:UTF-8
ENV LC_ALL en_GB.UTF-8

RUN apt-get update && apt-get install -y --no-install-recommends libncurses5 && \
    apt-get clean && rm -Rf /var/lib/apt/lists/*

ADD ./mdyalog /opt/mdyalog
ADD entrypoint /

RUN ln -s /run /usr/bin/dyalog

RUN useradd -s /bin/bash -d /home/dyalog -m dyalog

RUN mkdir /app /storage && \
    chmod 777 /app /storage

```

DYALOG 2020

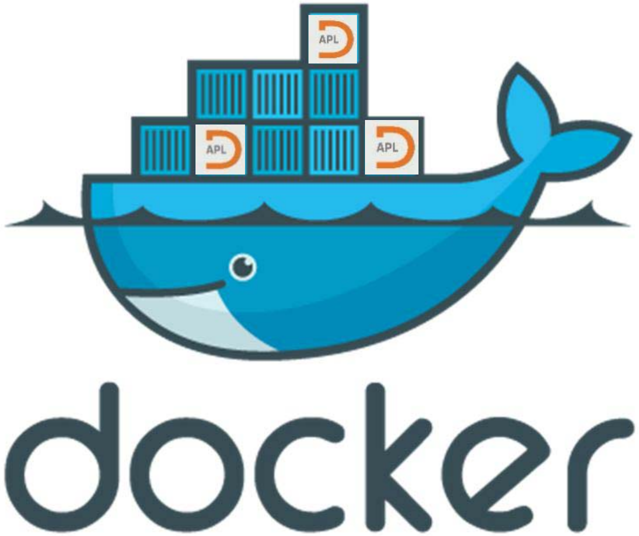
Recap: Dyalog's Public Containers

At https://hub.docker.com/u/dyalog

Image Name	Description	Boot Folder
dyalog/dyalog	Minimal Linux with Dyalog APL	/app
dyalog/jarvis	dyalog/dyalog with Jarvis installed	/code
dyalog/miserver	... with MiServer installed	/misite
dyalog/jupyter	For Dyalog-backed Jupyter notebooks	/app
dyalog/jsonserver	Use dyalog/jarvis instead	/code

Should we run a half-day workshop?

- Write to me if you would attend



Visit https://dyalog.tv

- Recordings of Dyalog'20 sessions will start to appear on Friday
- Hours of material on Dyalog v18.0 highlights
- A rapidly growing collection of presentations on the use of APL