OpenACS and EuroTcl 2024

NaviServer 5.0

Univ.-Prof. Dr. Gustaf Neumann

Vienna University of Economics and Business
Information Systems and New Media

JULY 8, 2024

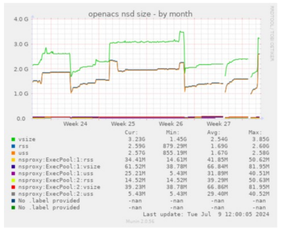
Overview

- What's new?
  - Part 1 of NaviServer 5 was presented here last year https://openacs.org/conf2023/info/download/file/openacs-conf-2023-naviserver.pdf
  - NaviServer 5 release Tcl 9 compatible
  - … depends on Tcl 9 release
  - … was infected with the Tcl 9 disease
  - Changes since last year:

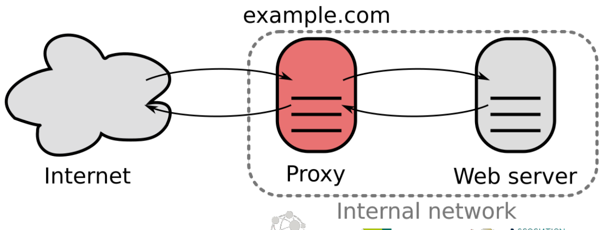
```
324 files changed, 16365 insertions(+), 7648 deletions(-)
```

- Most important developments since last year
  - Container support
  - Tracking memory growth
  - Handling large files
  - Unix Domain Sockets
  - NaviServer and Reverse Proxy Servers

- Next Steps



```
                                    Cur:      Min:      Avg:      Max:
■ vsize                            3.23G     1.45G     2.54G     3.85G
■ rss                              2.59G   879.29M     1.69G     2.60G
■ uss                              2.57G   855.19M     1.67G     2.58G
■ nsproxy:ExecPool:1:rss          34.41M    14.61M    41.85M    50.62M
■ nsproxy:ExecPool:1:vsize        61.52M    38.78M    66.84M    81.95M
■ nsproxy:ExecPool:1:uss          25.21M     5.43M    31.89M    40.51M
■ nsproxy:ExecPool:2:rss          14.52M    14.52M    39.29M    50.63M
■ nsproxy:ExecPool:2:vsize        39.23M    38.78M    66.86M    81.95M
■ nsproxy:ExecPool:2:uss           5.43M     5.43M    29.40M    40.52M
■ No .label provided                -nan      -nan      -nan      -nan
■ No .label provided                -nan      -nan      -nan      -nan
                                  Last update: Tue Jul  9 12:00:05 2024
```

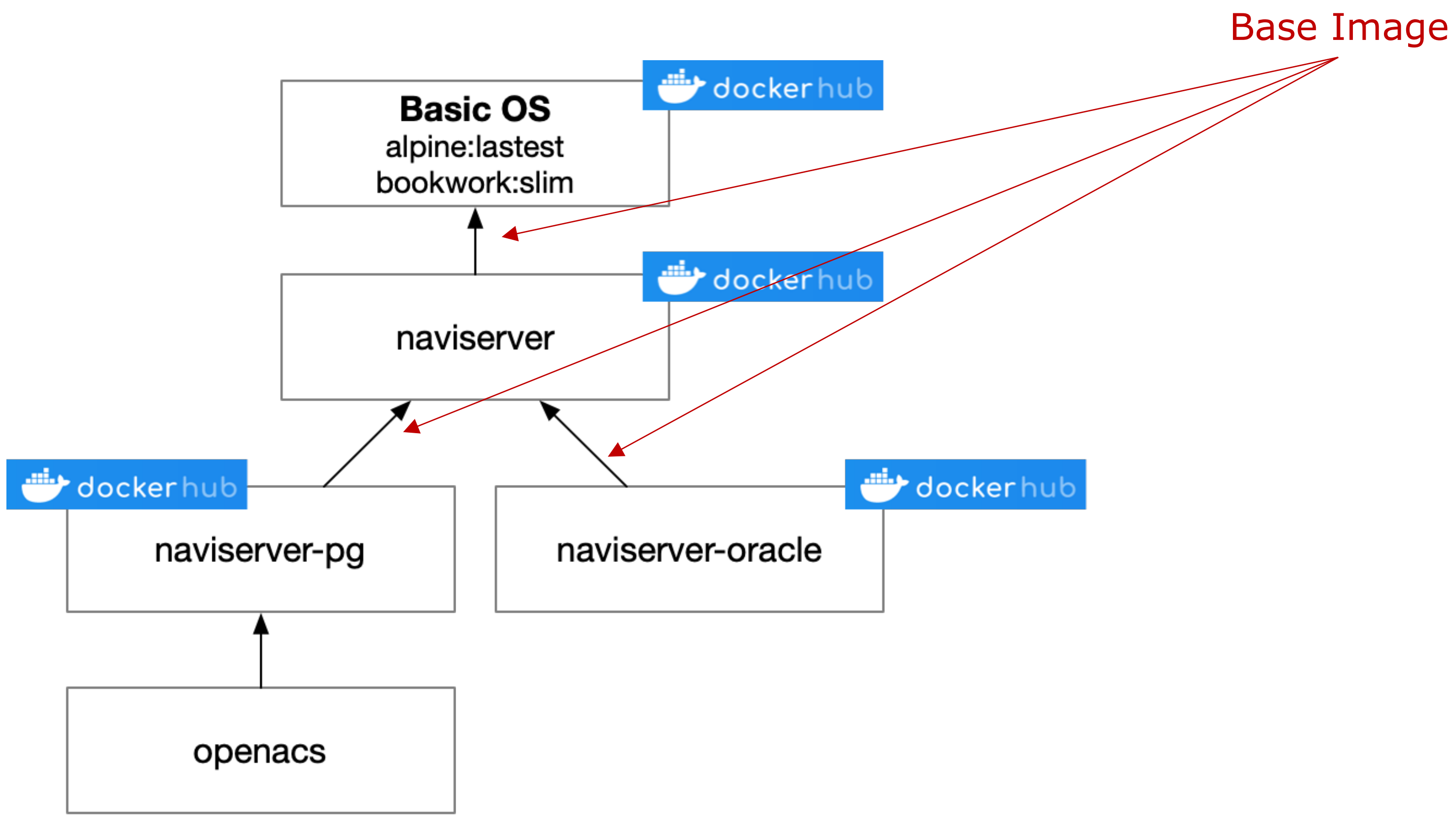


NaviServer in a Container

- Goals:
  - Ease deployment
  - Ease testing with multiple versions of NaviServer/Tcl/…
  - Based on Alpine and Debian
  - Binaries for linux/amd64 and linux/arm64
  - Preconfigured with most common NaviServer modules

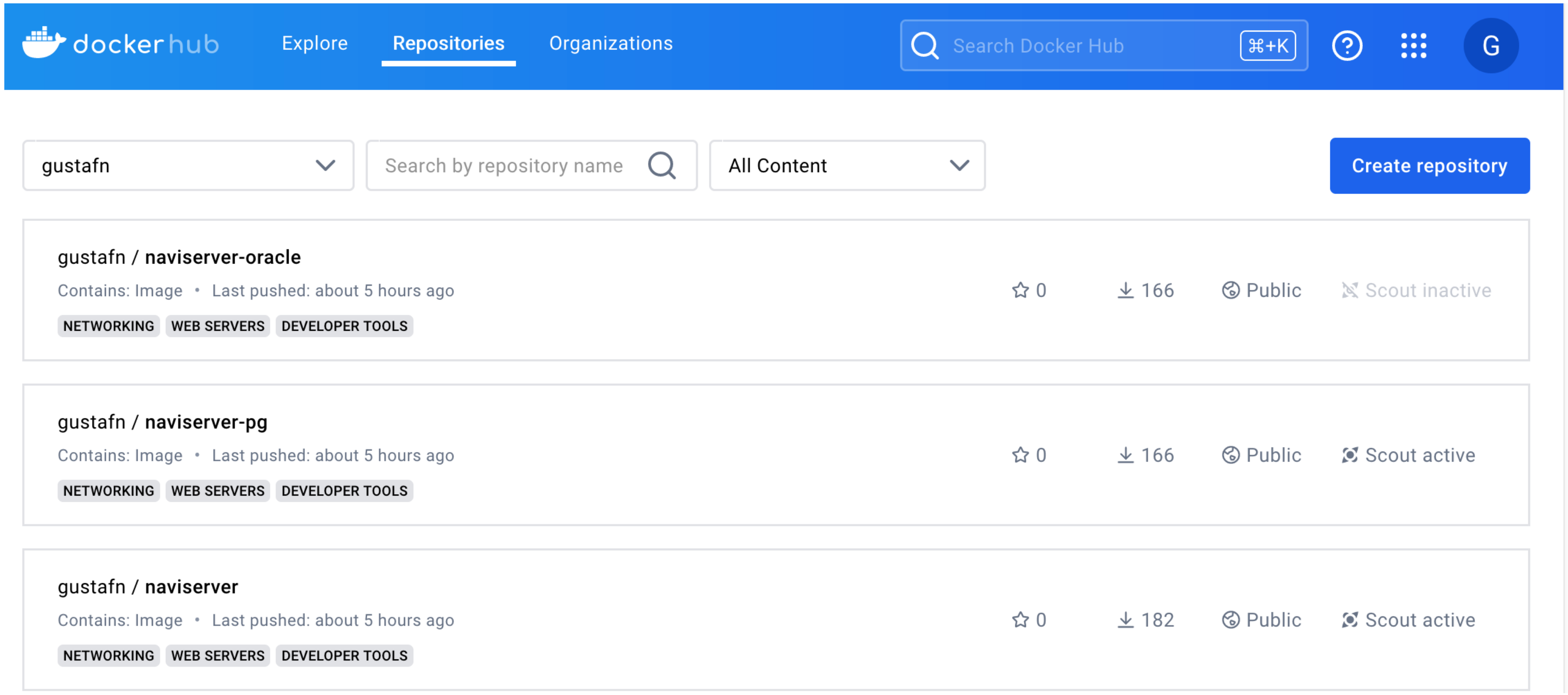
- Containers:
  - Base NaviServer
  - NaviServer with PostgreSQL client support
  - NaviServer with Oracle client support
  - OpenACS based on NaviServer with PostgreSQL client support

NaviServer Container Structure



Repositories at hub.docker.com

3 dockerhub repositories for NaviServer

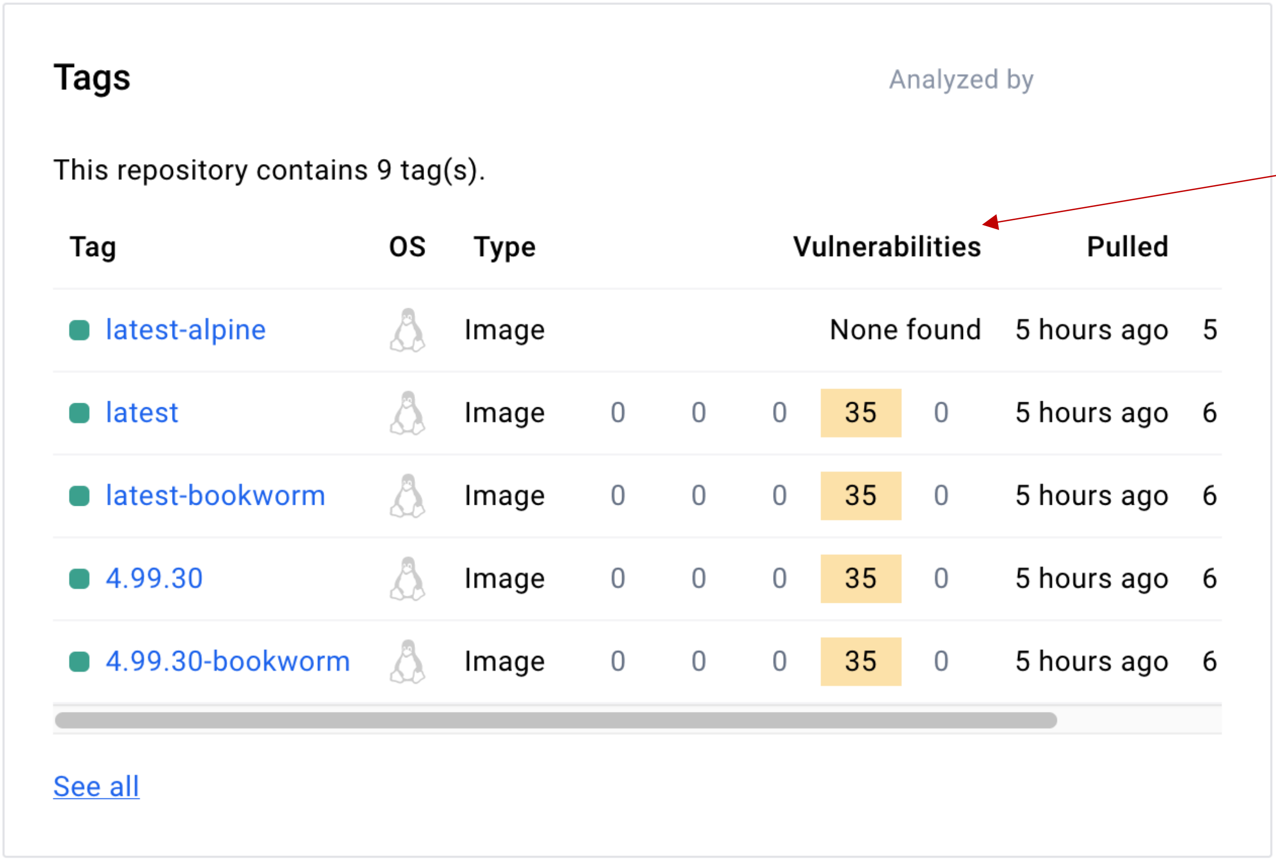


gustafn / **naviserver-oracle**
Contains: Image • Last pushed: about 5 hours ago
NETWORKING WEB SERVERS DEVELOPER TOOLS
☆ 0 ⤓ 166 Public Scout inactive

gustafn / **naviserver-pg**
Contains: Image • Last pushed: about 5 hours ago
NETWORKING WEB SERVERS DEVELOPER TOOLS
☆ 0 ⤓ 166 Public Scout active

gustafn / **naviserver**
Contains: Image • Last pushed: about 5 hours ago
NETWORKING WEB SERVERS DEVELOPER TOOLS
☆ 0 ⤓ 182 Public Scout active

Repository: gustafn/naviserver



Tags

Analyzed by

This repository contains 9 tag(s).

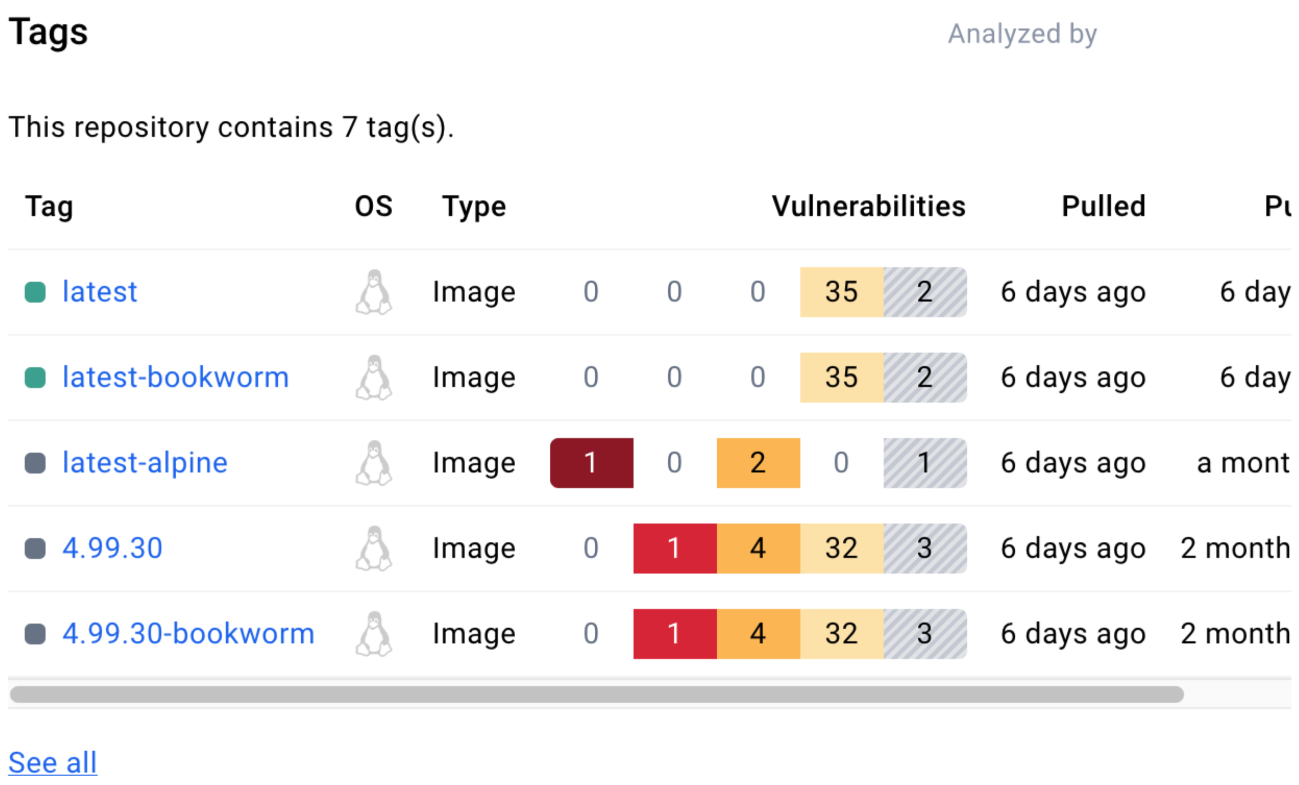
Tag	OS	Type				Vulnerabilities		Pulled	
🟩 latest-alpine		Image				None found		5 hours ago	5
🟩 latest		Image	0	0	0	35	0	5 hours ago	6
🟩 latest-bookworm		Image	0	0	0	35	0	5 hours ago	6
🟩 4.99.30		Image	0	0	0	35	0	5 hours ago	6
🟩 4.99.30-bookworm		Image	0	0	0	35	0	5 hours ago	6

See all

Image analysis
by Docker Scout

Per repository 9 variants (tags):
- latest, latest-bookworm, latest-alpine
- last releases: 4.99.30 + variants, 4.99.29 + variants

Repository: gustafn/naviserver (a few days ago)



Tags

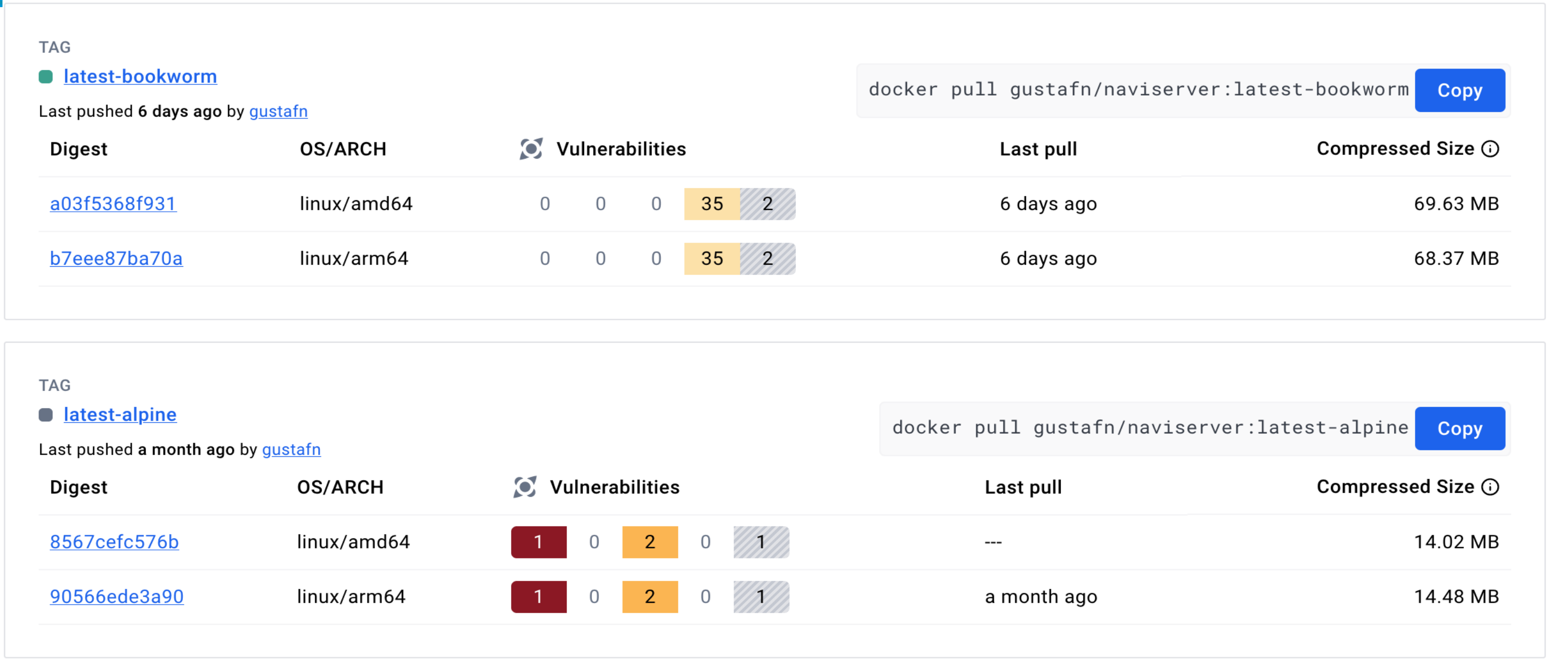
Analyzed by

This repository contains 7 tag(s).

Tag	OS	Type	Vulnerabilities					Pulled	Pu
🟩 latest	🐧	Image	0	0	0	35	2	6 days ago	6 day
🟩 latest-bookworm	🐧	Image	0	0	0	35	2	6 days ago	6 day
■ latest-alpine	🐧	Image	1	0	2	0	1	6 days ago	a mont
■ 4.99.30	🐧	Image	0	1	4	32	3	6 days ago	2 month
■ 4.99.30-bookworm	🐧	Image	0	1	4	32	3	6 days ago	2 month

See all

Per Tag: images for amd64 and arm64



TAG

🟩 latest-bookworm

Last pushed **6 days ago** by gustafn

```
docker pull gustafn/naviserver:latest-bookworm
```

Copy

Digest	OS/ARCH	Vulnerabilities					Last pull	Compressed Size ⓘ
a03f5368f931	linux/amd64	0	0	0	35	2	6 days ago	69.63 MB
b7eee87ba70a	linux/arm64	0	0	0	35	2	6 days ago	68.37 MB

TAG

● latest-alpine

Last pushed **a month ago** by gustafn

```
docker pull gustafn/naviserver:latest-alpine
```

Copy

Digest	OS/ARCH	Vulnerabilities					Last pull	Compressed Size ⓘ
8567cefc576b	linux/amd64	1	0	2	0	1	---	14.02 MB
90566ede3a90	linux/arm64	1	0	2	0	1	a month ago	14.48 MB

Small footprint (compressed)

- Bookworm slim: 70 MB (pg 90 MB, ora 150MB)
- Alpine: 14 MB (pg: 15MB)

Number of images

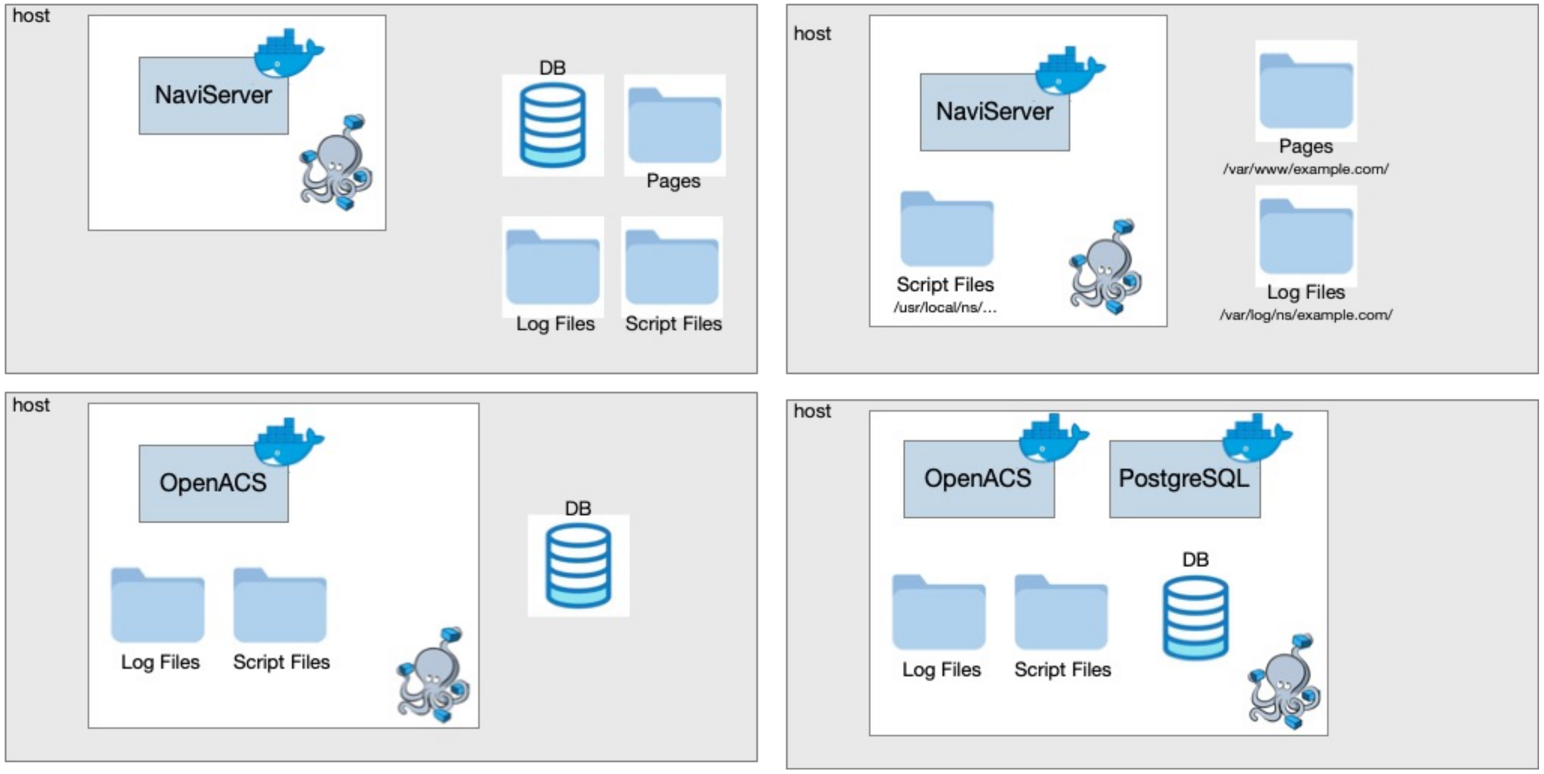
- REPOS * tags * images = 3 * 9 * 2 = 54

Why not always Alpine?

- Based on "musl"
- Currently no "`tcmalloc`"
- "`lc_collate`" limited
- Oracle client binary libraries don't work (arm64)

```
gustafn/naviserver latest-alpine  d654f2becdb1  14 minutes ago   53MB
```

Many Configuration Options
(What should be in the container?)



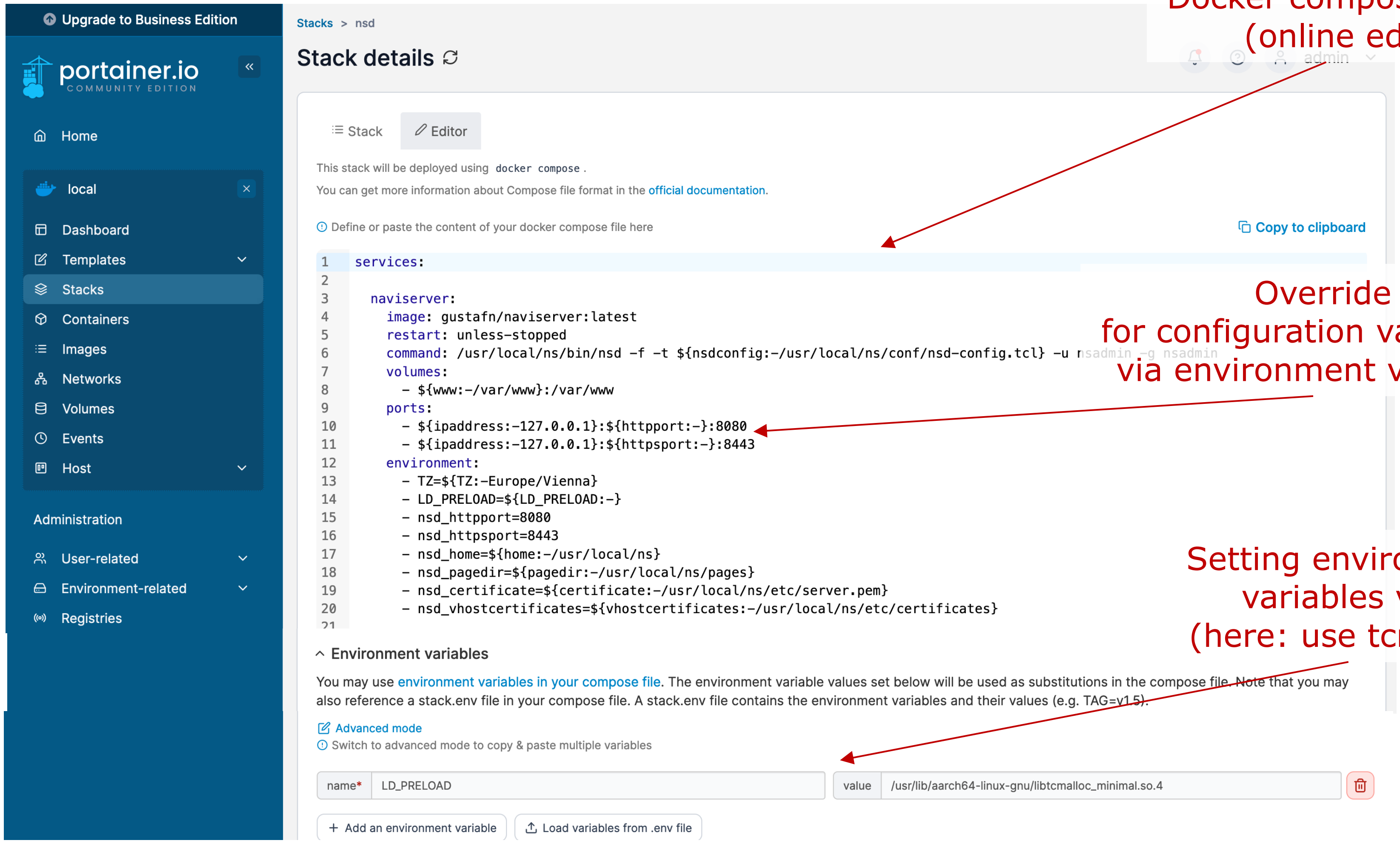
Options (sample)

- Container contains only binaries
- Container contains binaries and script files
- Keep log-files and script files in container, …
- Access DB on host
- Run DB and NaviServer in container

Tool of choice

- Docker compose
- All these variants can be defined in a single file
- Avoid producing multiple config files
- Configurations are “stacks”

Recommendation: Use docker compose and Portainer



This stack will be deployed using `docker compose`.

You can get more information about Compose file format in the official documentation.

Define or paste the content of your docker compose file here

Copy to clipboard

```
services:

  naviserver:
    image: gustafn/naviserver:latest
    restart: unless-stopped
    command: /usr/local/ns/bin/nsd -f -t ${nsdconfig:-/usr/local/ns/conf/nsd-config.tcl} -u nsadmin -g nsadmin
    volumes:
      - ${www:-/var/www}:/var/www
    ports:
      - ${ipaddress:-127.0.0.1}:${httpport:-}:8080
      - ${ipaddress:-127.0.0.1}:${httpsport:-}:8443
    environment:
      - TZ=${TZ:-Europe/Vienna}
      - LD_PRELOAD=${LD_PRELOAD:-}
      - nsd_httpport=8080
      - nsd_httpsport=8443
      - nsd_home=${home:-/usr/local/ns}
      - nsd_pagedir=${pagedir:-/usr/local/ns/pages}
      - nsd_certificate=${certificate:-/usr/local/ns/etc/server.pem}
      - nsd_vhostcertificates=${vhostcertificates:-/usr/local/ns/etc/certificates}
```

Environment variables

You may use environment variables in your compose file. The environment variable values set below will be used as substitutions in the compose file. Note that you may also reference a stack.env file in your compose file. A stack.env file contains the environment variables and their values (e.g. TAG=v1.5).

Advanced mode

Switch to advanced mode to copy & paste multiple variables

name*	LD_PRELOAD	value	/usr/lib/aarch64-linux-gnu/libtcmalloc_minimal.so.4

+ Add an environment variable | Load variables from .env file

Docker compose file (online editing)

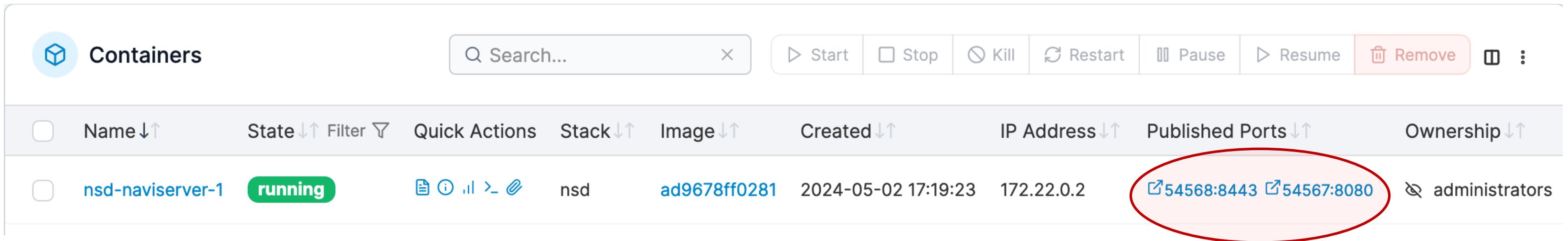
Override defaults for configuration variables via environment variables

Setting environment variables via GUI (here: use tcmalloc)

Networking Challenges with Containers (1/2)

Server-side Ephemeral Ports

- “short living” ports
- Usually used for client side
- Container context: used for servers to ease start of multiple servers of the same kind
    - running on docker host multiple nsd
    - every container has a different external listening port)
- Of course, server port does not have to be ephemeral



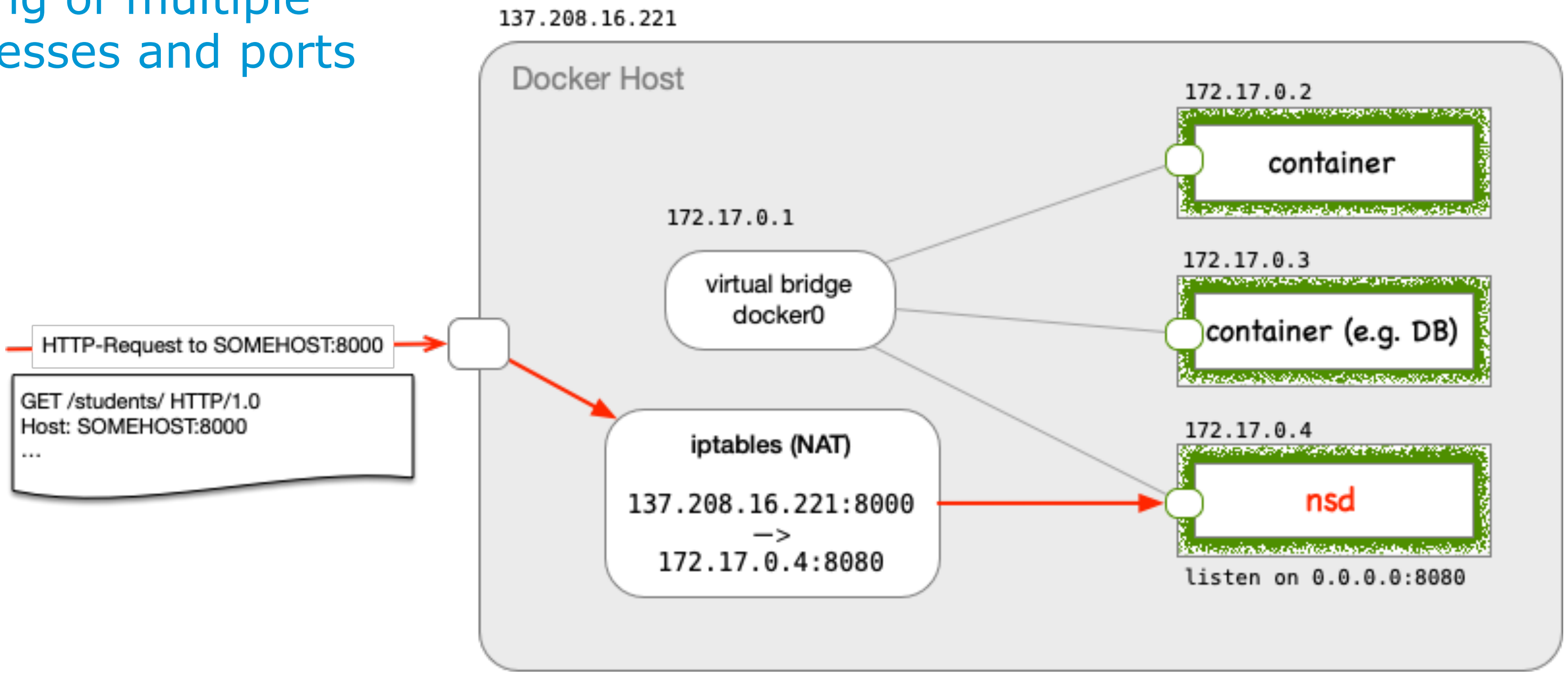
Name	State	Quick Actions	Stack	Image	Created	IP Address	Published Ports	Ownership
nsd-naviserver-1	running		nsd	ad9678ff0281	2024-05-02 17:19:23	172.22.0.2	54568:8443 54567:8080	administrators

54567:8080 54568:8443

Mapping of
- port from docker host to
- listening port in container

Networking Challenges with Containers (2/2)

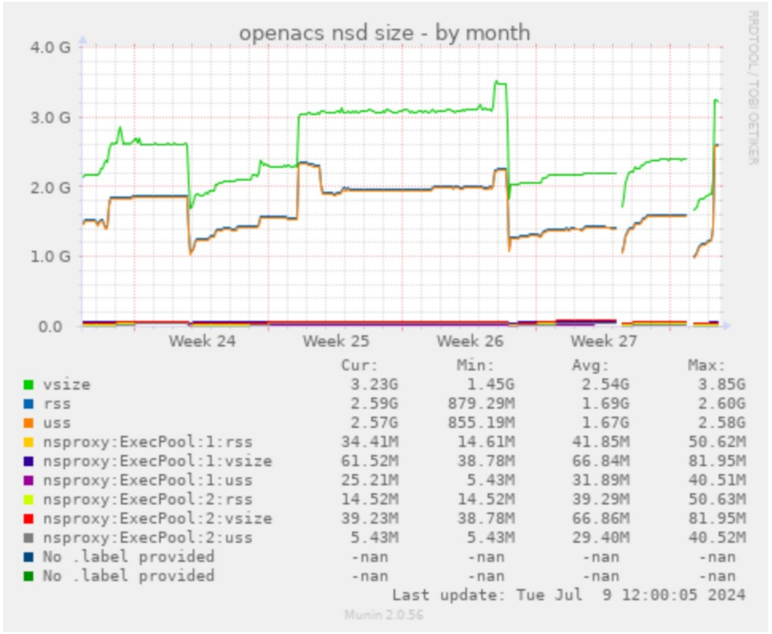
Managing of multiple IP addresses and ports



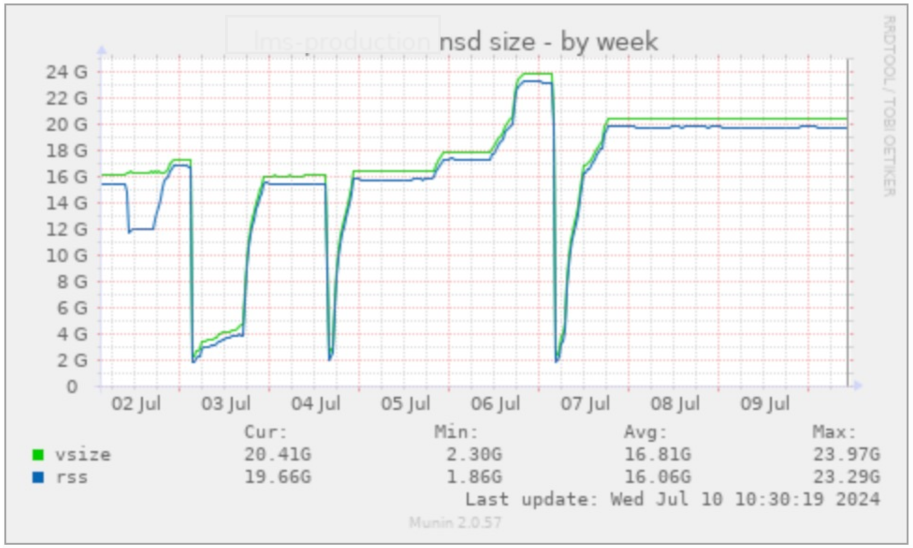
Challenges inside the container

- Validation of host header fields
- Addressing DB e.g. on docker host or in a different container
- Telling other servers in, e.g., an OpenACS cluster your “external” address (defined in docker-compose file, ephemeral ports)
- Some HTTP requests should run solely inside the container (e.g. regression test)
- … lead to changes in OpenACS

Understanding Memory Growth (1/3)



```
                                 Cur:      Min:      Avg:      Max:
■ vsize                         3.23G     1.45G     2.54G     3.85G
■ rss                           2.59G   879.29M     1.69G     2.60G
■ uss                           2.57G   855.19M     1.67G     2.58G
■ nsproxy:ExecPool:1:rss       34.41M    14.61M    41.85M    50.62M
■ nsproxy:ExecPool:1:vsize     61.52M    38.78M    66.84M    81.95M
■ nsproxy:ExecPool:1:uss       25.21M     5.43M    31.89M    40.51M
■ nsproxy:ExecPool:2:rss       14.52M    14.52M    39.29M    50.63M
■ nsproxy:ExecPool:2:vsize     39.23M    38.78M    66.86M    81.95M
■ nsproxy:ExecPool:2:uss        5.43M     5.43M    29.40M    40.52M
■ No .label provided             -nan      -nan      -nan      -nan
■ No .label provided             -nan      -nan      -nan      -nan
                                Last update: Tue Jul  9 12:00:05 2024
                              Munin 2.0.56
```



```
              Cur:          Min:          Avg:          Max:
■ vsize      20.41G         2.30G        16.81G        23.97G
■ rss        19.66G         1.86G        16.06G        23.29G
                          Last update: Wed Jul 10 10:30:19 2024
                  Munin 2.0.57
```

Many potential reasons

- Configuration specific (increasing limits dynamically)
- Cache growth
- Application leaks (namespaced variables are not cleaned up after request)
- C-level leaks from NaviServer and/or modules (using valgrind)
- Tcl (doubling policy for space in Tcl_Obj, Tcl_DString, …, might survive long)
- OS + C-library (esp. malloc implementation)
- Memory fragmentation

Background multipart/form-data

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

HTML Form

```
<FORM action="/cgi/handle"
           enctype="multipart/form-data"
           method="POST">
  What is your name? <INPUT type=”text" name="submitter">
  What files are you sending? <INPUT type =”file" name ="pics">
<FORM>
```

HTTP Request

Request data provided by NaviServer in memory (when small) or via spool file (when large)

```
POST /upload HTTP/1.1
Content-Length: 14128
Content-type: multipart/form-data, boundary=AaB03x

--AaB03x
content-disposition: form-data; name=”submitter"

Susie Derkins
--AaB03x
content-disposition: form-data; name="pics"; filename="file1.txt"
Content-Type: text/plain

... contents of file1.txt ...
--AaB03x--
```

Understanding Memory Growth (2/3)

Snippet from parsing a file containing `multipart/formdata` in Tcl

```
#
# Read lines of data until another boundary is found.
#
while { ![eof $fp] } {
    if { [string match $boundary* [string trim [gets $fp]]] } {
        break
    }
    set end [tell $fp]
}
set length [expr {$end - $start - 2}]
```

This snippet can cause memory bloat and/or crash!

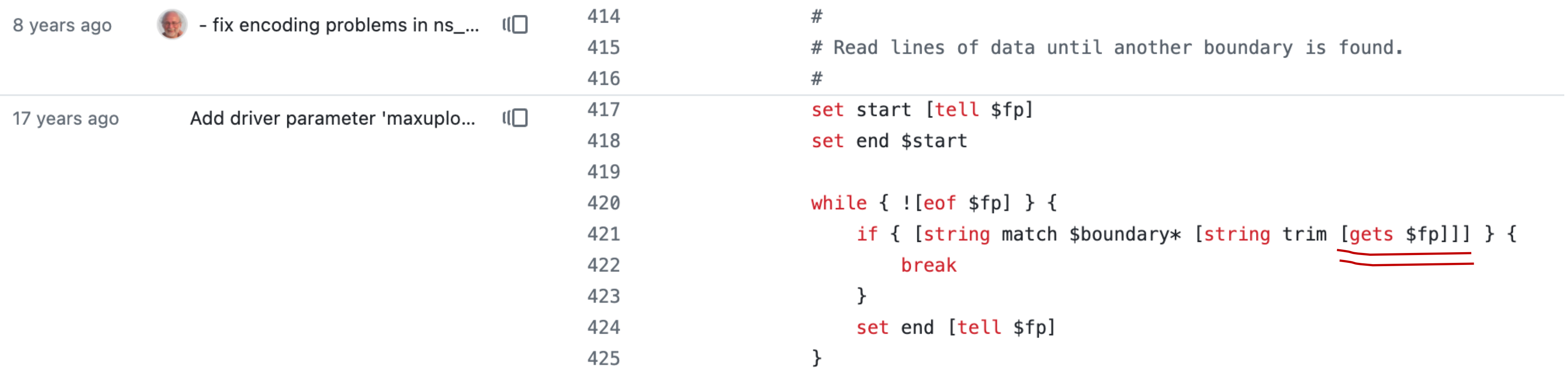
Can you spot it?

Abbreviated sample request data spooled to a file

```
POST /upload HTTP/1.1
Content-Length: 14128
Content-type: multipart/form-data, boundary=AaB03x

--AaB03x
content-disposition: form-data; name="submitter"

Susie Derkins
--AaB03x
content-disposition: form-data; name="pics"; filename="file1.txt"
Content-Type: text/plain

... contents of file1.txt ...
--AaB03x--
```

Understanding Memory Growth (2/3)

Snippet “worked” for 17 years



8 years ago - fix encoding problems in ns_...

17 years ago Add driver parameter 'maxuplo...

```
        #
        # Read lines of data until another boundary is found.
        #
        set start [tell $fp]
        set end $start

        while { ![eof $fp] } {
            if { [string match $boundary* [string trim [gets $fp]]] } {
                break
            }
            set end [tell $fp]
        }
```

Tcl “gets” potentially harmful

- Result of `gets` is a Tcl_Obj
- Can cause a memory bloat/crash, when newlines are more than 2 GB apart
- Code was unchanged at least for 17 years
- Was found when testing known issues with Tcl 9.
- With Tcl 9 the crash disappeared, but the memory bloat stays
- New solution in NaviServer 5: `ns_fseekchars`

Parsing large file uploads (multipart/formdata) in Tcl using `ns_fseekchars`

```
#
# Read lines of data until another boundary is found.
#
seek $fp $start
set t1 [time {
    set seekChar [ns_fseekchars $fp \n$boundary]
    if {$seekChar == -1} {
        error "boundary not found"
    }
    # move beyond the leading newline
    incr seekChar
    set end    [expr {$seekChar - [string length $boundary]}]
    set length [expr {$end - $start - 2}]
}]
```

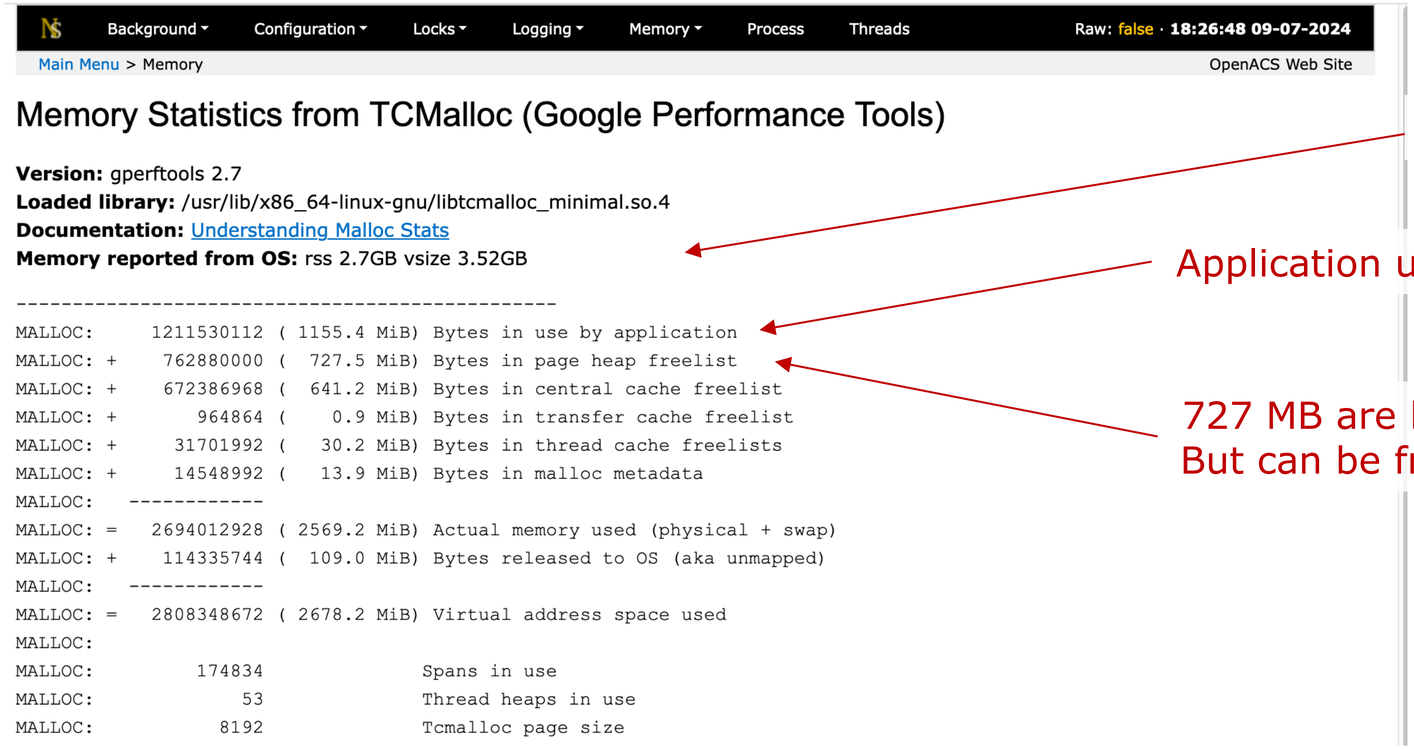
microseconds

```
    file size         old  ns_fseekchars  factor
       65,517       4,471            151   29.61
      124,523       1,139             94   12.12
   74,006,378     682,375         54,752   12.46
2,104,408,064  18,916,496      1,564,472   12.09
3,992,977,408  35,942,768      3,061,061   11.74
5,368,709,120                  3,817,896
```

Benefits:

- No memory bloat
- Lifts 4 GB limit for Tcl 8.6 as well
- Significantly faster

Understanding Memory Growth (3/3)



Memory Statistics from TCMalloc (Google Performance Tools)

Version: gperftools 2.7
Loaded library: /usr/lib/x86_64-linux-gnu/libtcmalloc_minimal.so.4
Documentation: Understanding Malloc Stats
Memory reported from OS: rss 2.7GB vsize 3.52GB

```
------------------------------------------------
MALLOC:     1211530112 ( 1155.4 MiB) Bytes in use by application
MALLOC: +    762880000 (  727.5 MiB) Bytes in page heap freelist
MALLOC: +    672386968 (  641.2 MiB) Bytes in central cache freelist
MALLOC: +       964864 (    0.9 MiB) Bytes in transfer cache freelist
MALLOC: +     31701992 (   30.2 MiB) Bytes in thread cache freelists
MALLOC: +     14548992 (   13.9 MiB) Bytes in malloc metadata
MALLOC:   ------------
MALLOC: =   2694012928 ( 2569.2 MiB) Actual memory used (physical + swap)
MALLOC: +    114335744 (  109.0 MiB) Bytes released to OS (aka unmapped)
MALLOC:   ------------
MALLOC: =   2808348672 ( 2678.2 MiB) Virtual address space used
MALLOC:
MALLOC:         174834              Spans in use
MALLOC:             53              Thread heaps in use
MALLOC:           8192              Tcmalloc page size
```

RSS: 2.7 GB

Application usage: 1.1 GB

727 MB are kept for reuse,
But can be freed (via Web)

Newly integrated statistics from TCmalloc

- Part of "nsstats"
- Requires compilation with –DSYSTEM_MALLOC for Tcl and NaviServer (or flag in install-ns)
- Requires setting LD_PRELOAD (see above)
- Comparison of malloc implementations https://next-scripting.org/2.4.0/doc/misc/thread-mallocs

Unix Domain Sockets

- Motivation
  - Reduce networking complexity (e.g. with containers)
  - Uses Unix permission system (access control, easier than firewall)
  - Probably better performance (low latency, high throughput)
  - Resource efficiency (no networking stack involved)
  - User request (Georg asked)
- Unix Domain Socket Support in NaviServer 5:
  - Incoming requests (server side)
  - Outgoing requests (client side)
    - `ns_http`
    - `ns_connchan`

Unix Domain Sockets (Server Side)

- How
  - Implemented via the `nssock` module (general socket implementation)
  - Parameter `address` must start with a "/", no `port` needed
  - Unix Domain Socket is created upon server start

```
#
# Example driver configuration for listening on a Unix Domain Socket
#
ns_section ns/modules {
    ns_param unix nssock
}

ns_section ns/module/unix {
    ns_param defaultserver  default
    ns_param address        /tmp/uds.socket
 }
```

Unix Domain Sockets (Client Side)

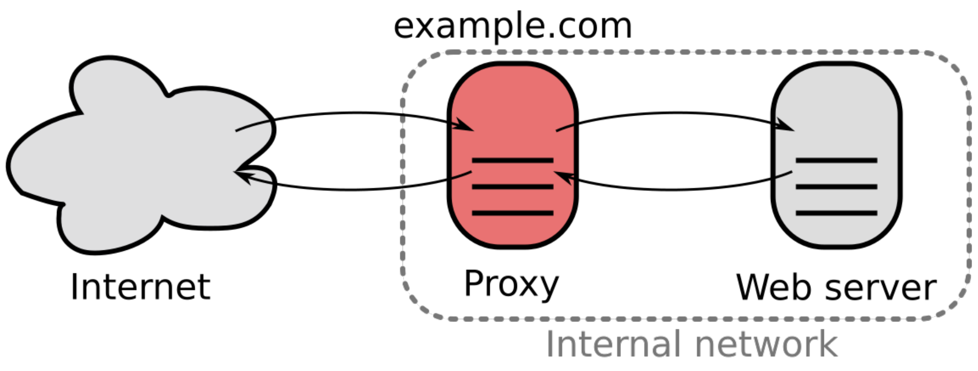
- How
  - Added flag `-unix_socket SOCKETNAME` to `ns_http` and `ns_connchan` (similar cURL)

```
#
# Example of using ns_connchan via domain socket
#
set chan [ns_connchan open -unix_socket /tmp/uds.socket http://foo.org/]
ns_connchan read $chan


#
# Example of using ns_nttp via domain socket
#
set d [ns_http run -unix_socket /tmp/uds.socket http://foo.org/]
```

- For reverse proxy implementation (revproxy module, Apache syntax):

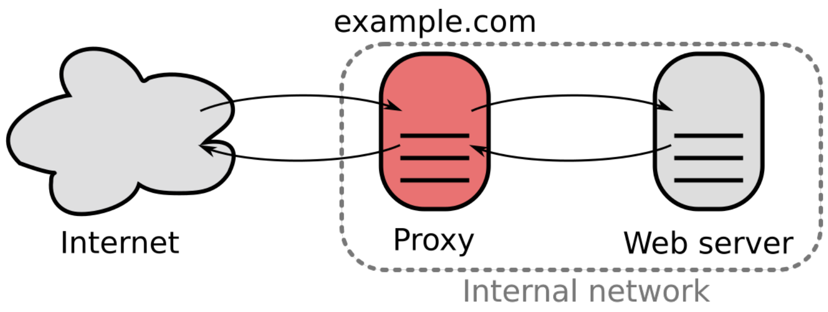
```
unix:/home/www.socket|http://localhost/whatever/
```

NaviServer as Reverse Proxy Server

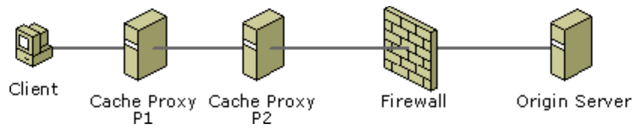


- Background
  - Implemented as NaviServer module
  - In use e.g. on openacs.org with virtual server cvs.openacs.org to redirect requests to “fisheye” server
  - Can be used as `filter` or via `ns_register_proc`
  - One can say redirects certain requests based on path or file name pattern to a different server
- New Features
  - Choice between `ns_connchan` and `ns_http`
  - Implementation based on `ns_http` can use persistent connections
  - Support of Unix Domain Sockets
  - One can now run OpenACS behind a NaviServer running as reverse proxy

NaviServer behind Reverse Proxy Server



- Challenge
  - Determine trusted peer addresses (who made the request) when running a reverse proxy
  - The raw peer address of the socket connection is always the proxy server
  - Peer IP addresses are needed in the log-files, access control, request queue management, trouble shooting, geo-location, …
  - Use of reverse proxies is growing (cloud, AWS load balancer, …)



- Standard Approach: x-forwarded-for, forward
  - Can be easily faked by a client adding its own content to x-forwarded-for
  - Complication: multi-tiered reverse proxies

```
X-Forwarded-For: client, proxy1, proxy2
```

- New
  - Configurable right-to-left processing (similar to the optional "realip" modules for nginx), define trusted forwarded-for server via CIDR specs, etc.
  - Implementation of new commands `ns_ip public` or `ns_ip trusted` https://naviserver.sourceforge.io/5.0/naviserver/files/ns_ip.html

Summary

- NaviServer 5
  - Runs its regression test regularly with actual Tcl9 versions
  - Should be released when Tcl9 is finally released
  - Many new features (a few covered here)
- Agenda
  - Provide tagging scheme for docker including Tcl9
  - OpenACS 5.10.1 release
  - Then porting 5.10.1 to Tcl9 (release packages larger than tcllib)

```
-------------------------------------------------------------------
Language              files          blank        comment         code
-------------------------------------------------------------------
…
Tcl/Tk                 2315          66944          65858       320853
SQL                    1846          48461          46288       215038
…
```

  - Tcl9 migration tool (based on nagelfar) does not work for OpenACS (`adp_proc`, argument checking, …)
  - Porting NaviServer documentation?
- Questions?

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Institute for Information Systems and New Media
Welthandelsplatz 1, 1020 Vienna, Austria

UNIV.PROF. DR. Gustaf Neumann

T +43-1-313 36-4671
Gustaf.neumann@wu.ac.at
www.wu.ac.at

EQUIS ACCREDITED AACSB ACCREDITED AMBA ACCREDITED