



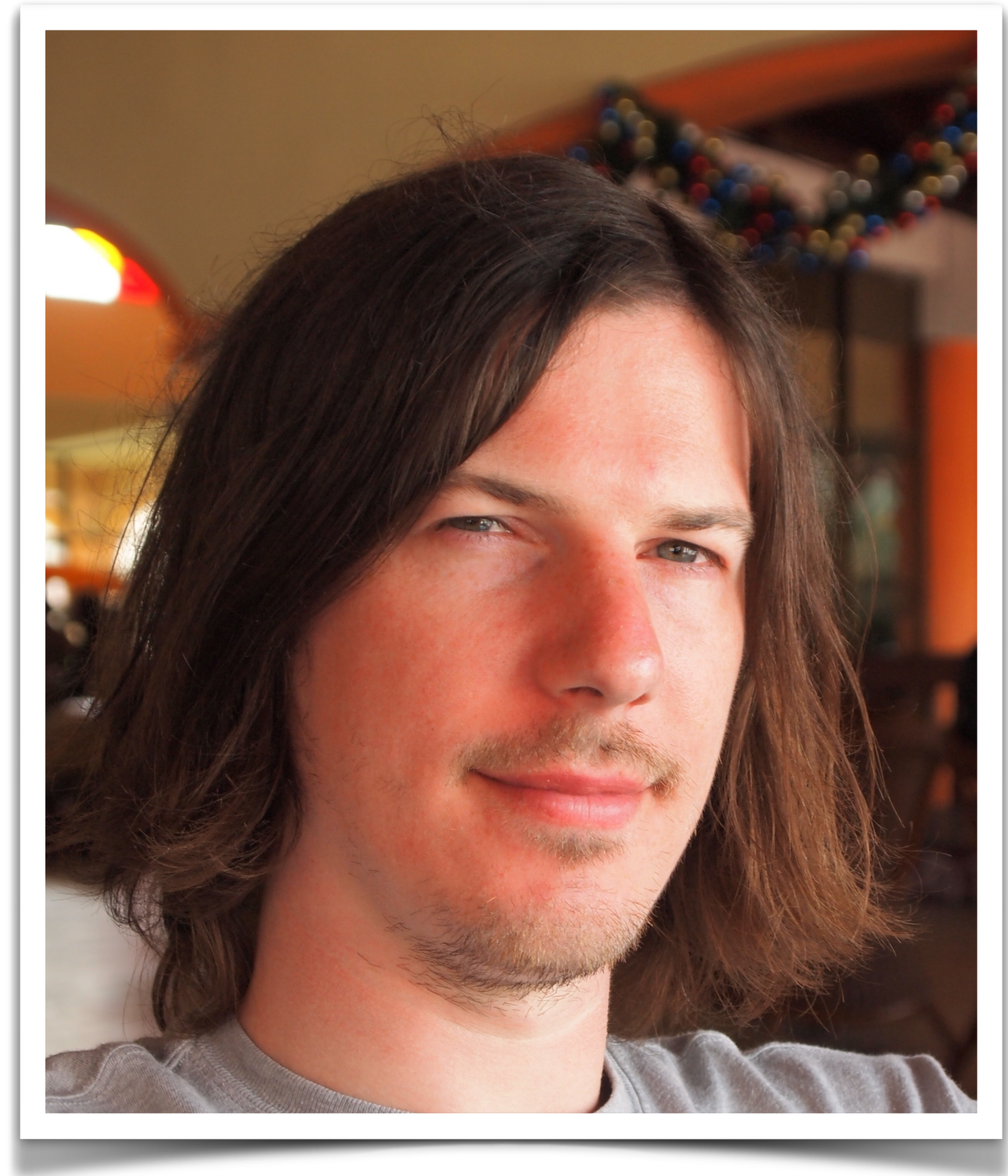
Replacing NSClient++ for Windows Monitoring

Sven Nierlein

09.11.2023

whoami

- Name: Sven Nierlein
- E-Mail: sven@consol.de
- Location: Munich, Germany
- Work: consol.de
- We might have met before here in Nuremberg:
 - 2022: Thruk
 - 2021: Naemon password vault
 - 2019: Naemon
 - 2018: Thruk
 - 2013: Thruk
 - 2012: Mod-Gearman

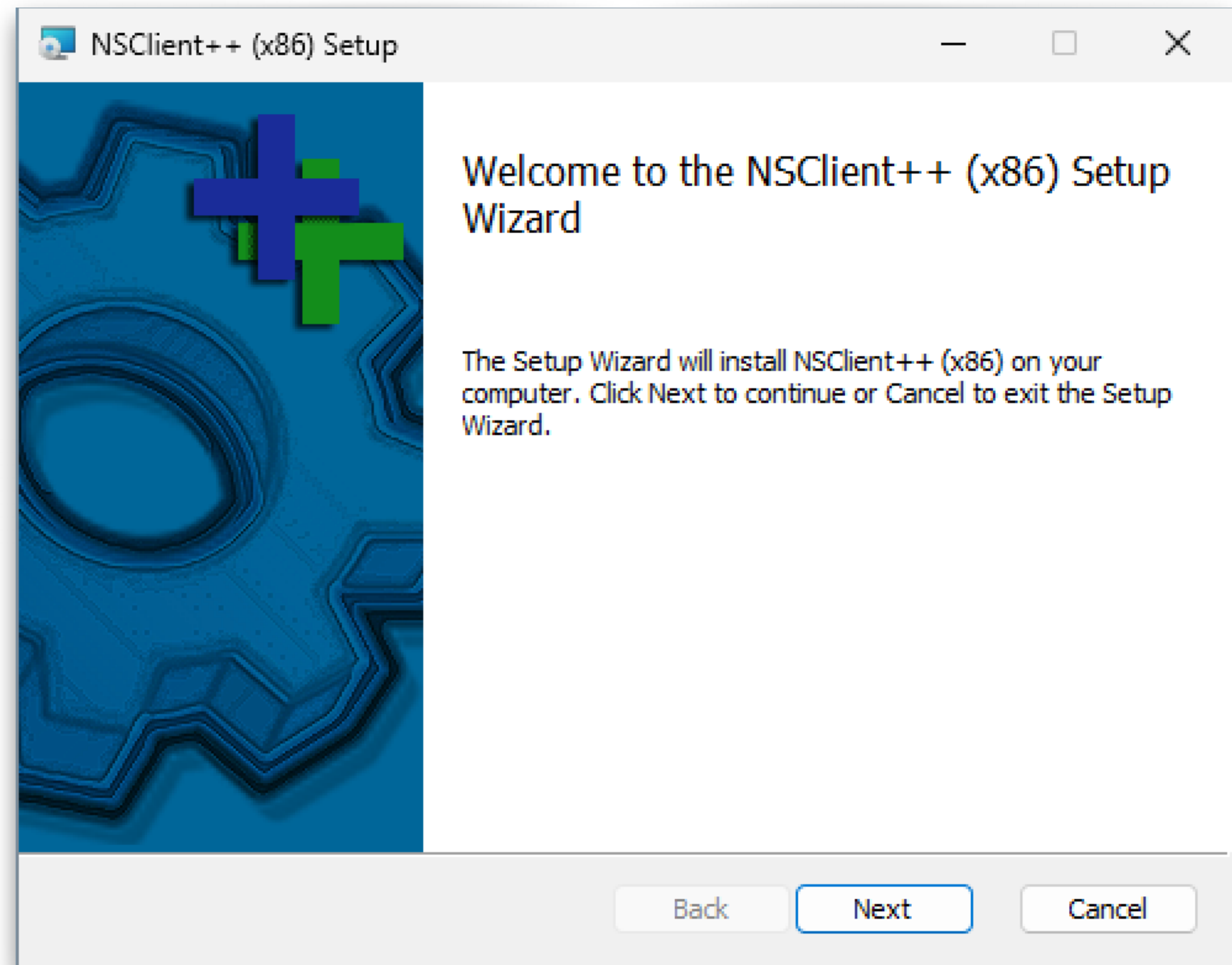




Introduction

Recap: NSClient++

- Windows Agent (and Linux)
- Supports several protocols
 - NRPE
 - REST
 - and more ...
- Lots of built-in checks
- Extendable
- Customisable



Status Quo

- Basically all our customers use NSClient++ in some way
- Largest setup ~15k 250k installations
- Usually happy with NSClient++



NSClient++ - So what's the issue?

- TLS <= 1.2
- Last release was years ago

```
Testing protocols via  
SSLv2      not offered  
SSLv3      not offered  
TLS 1      not offered  
TLS 1.1    offered (C  
TLS 1.2    offered (C  
TLS 1.3    not offered  
NPN/SPDY   not offered  
ALPN/HTTP2 not offered
```



Releases 128



0.5.2.41

Latest

on Apr 27, 2018

NSClient++

- But besides that, it just works...
- Anyway, we needed a solution for the SSL issue...



Going to

- Just up

	mickem added 30 commits <u>last year</u>	
	 Added some packages	ad76daf
	 hmmm	19a4832
	 ...	33d261f
	 q	2658969
	 ffff	542edfc
	 Bumped minim cmake version to remove a lot of errors	86831b6
	 random stuff from several months ago :)	a119365
	 hmmm	ad11960
	 grrr	833c9d4
	 *sigh*	1791e03
	 more fixes	0a3de82
	 hmmm	9b641c5
	 ...	9ff4499
	 Swapped to python 3 cmake modules	ec86960
	 Fixed a bunch of compioation issues	ac52e6f
	 strange	a6cdc95

Looking for alternatives

- There are a couple of options:
 - <https://github.com/infraweavers/monitoring-agent>
 - <https://github.com/continentale/sma-monitoring-agent/>
 - <https://github.com/continentale/monitoring-agent>
 - <https://github.com/it-novum/openitcockpit-agent-go>
 - <https://github.com/Icinga/icinga-powershell-framework>
 - Probably missed some more...



Alternative by [Nick Youngson](#) CC BY-SA 3.0 [Alpha Stock Images](#)

Requirements

- Drop-In Replacement
 - Support nsclient.ini Format
 - Must work on Windows and Linux
 - Modern SSL/TLS
 - Checks format/arguments compatible to NSClient
 - No Change to Naemon configuration
- Bonus Points
 - Prometheus





SNClient+

Introducing SNClient+

- Rewrite NSClient++ in Go
- Compatible with nsclient.ini
- Multiplatform
 - Windows
 - Linux
 - OSX, FreeBSD...
- Dual Stack
 - NSClient++
 - Prometheus

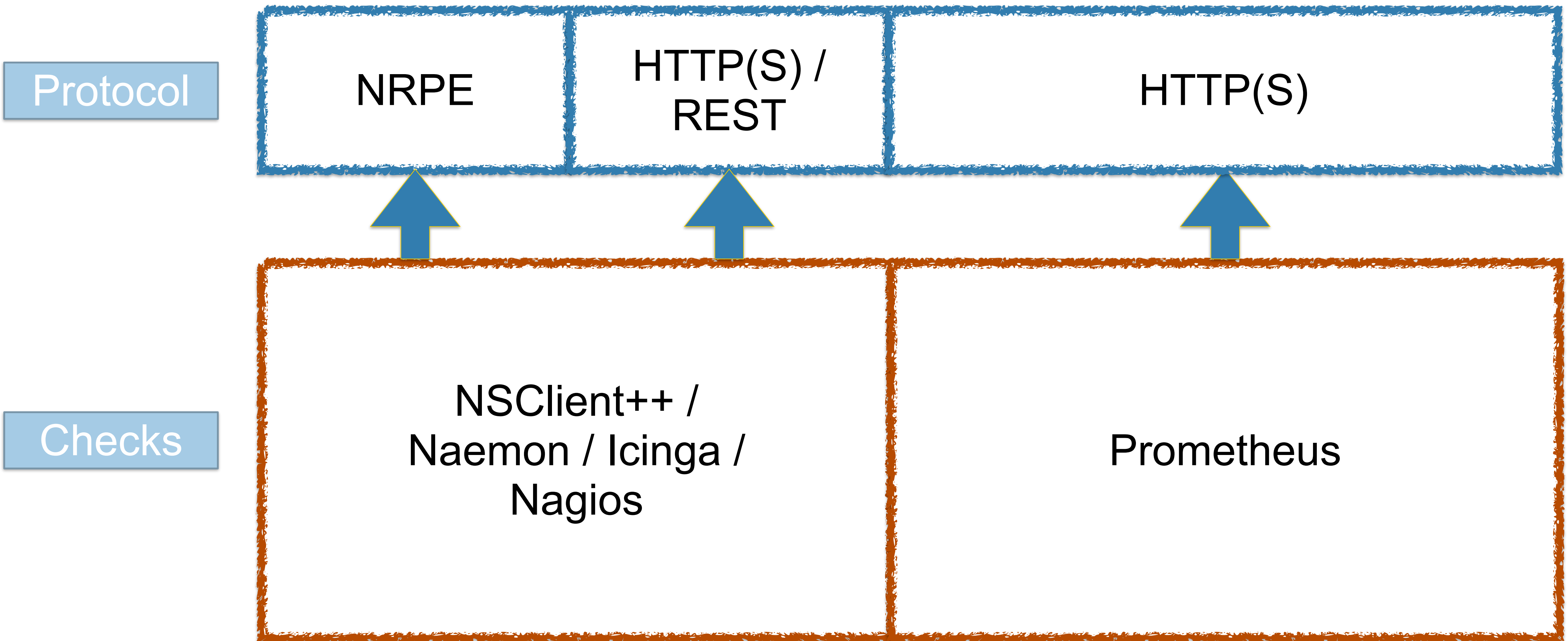


What does SNClient mean?

- Let's ask the chatbot of your choice:
 - Security/Networking Client
 - Software Name Client
 - Server Ninja Client
- The actual meaning is:
 - Secure Naemon Client



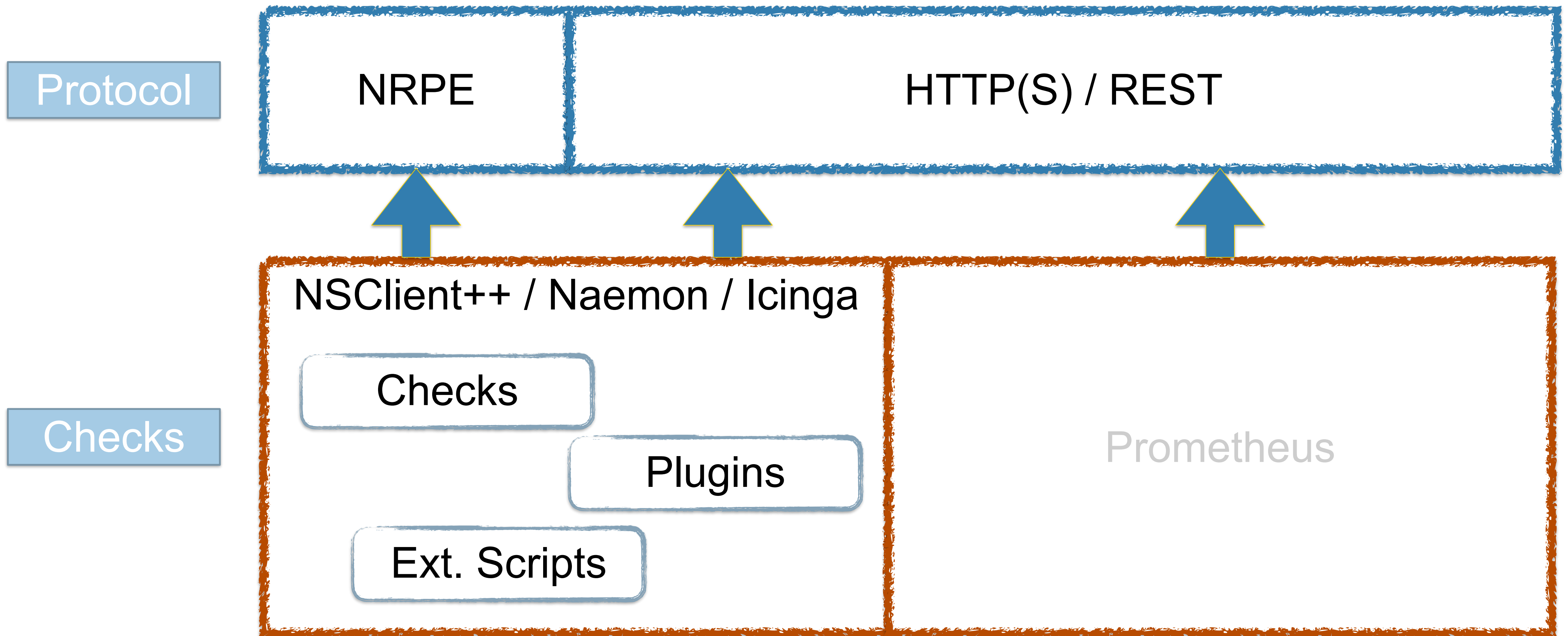
SNClient Dual-Stack





Naemon Stack

NSClient / Naemon Stack



SNClient

thruk

General

Home

Documentation

Panorama View

Current Status

Tactical Overview

Hosts

Services

Host Groups

Service Groups

Mine Map

Problems

search...

Reports

System

Current Network Status

Not Secure | https://192.168.208.2/demo/thruk/cgi-bin/status.cgi?style=detail

search...

Service Status Details

Hosts Services Problems In Downtime Unselect all

Display Filter 1 100 per page

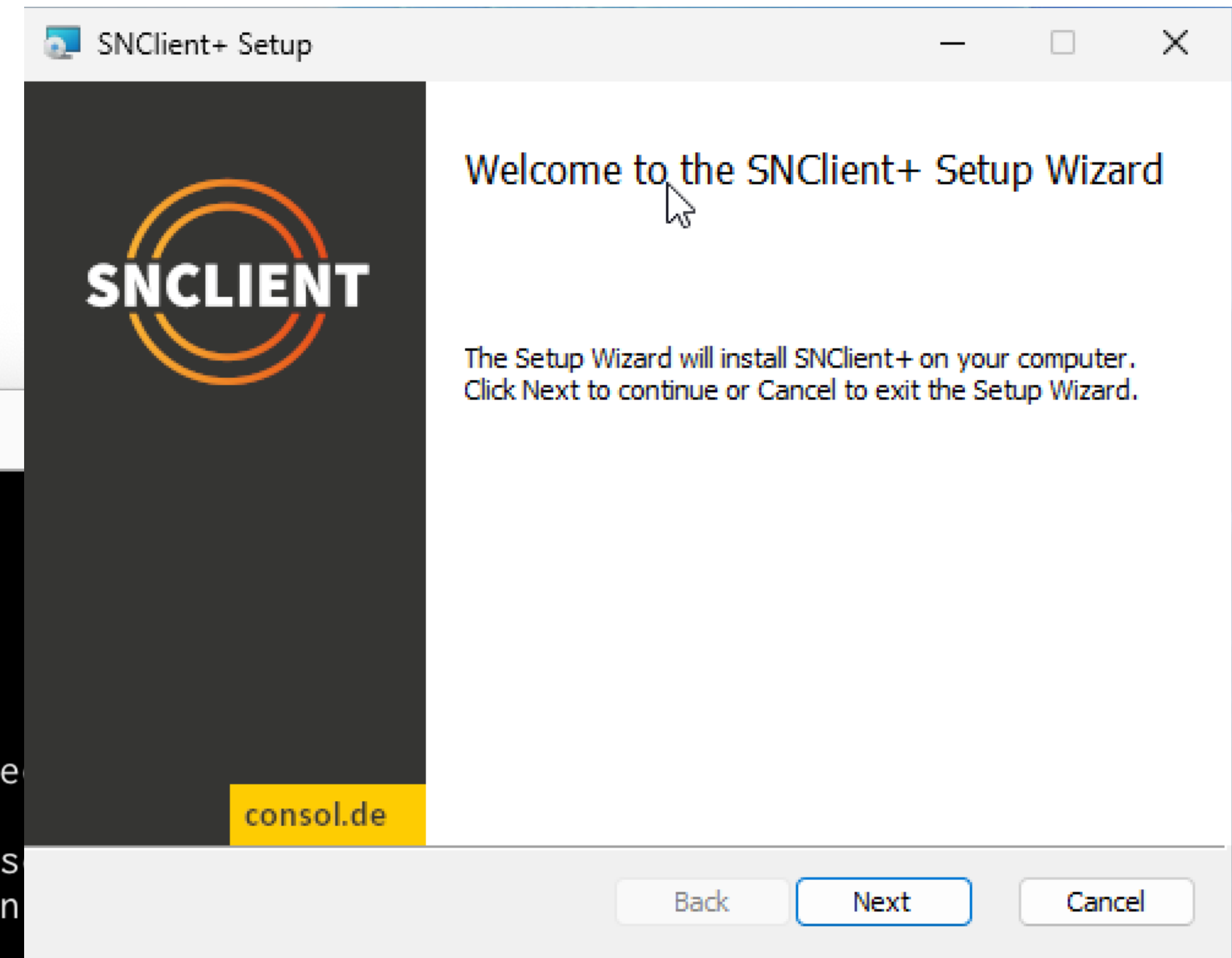
Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	agent inventory	PENDING	never	2s+	1/3	Service check scheduled for Tue Nov 7 14:45:44 CET 20...
	agent version	OK	14:09:18	1m 43s	1/3	SNClient+ v0.11.0023 (Build: cc5501f)
	cpu	OK	14:09:31	1m 30s	1/3	OK: CPU load is ok.
	disk /	OK	14:09:30	1m 31s	1/3	OK: All 1 drive(s) are ok
	memory	OK	14:09:55	1m 6s	1/3	OK: committed = 42.25 MiB, physical = 664...
	net enp0s1	OK	14:09:15	1m 46s	1/3	OK: enp0s1 >22569820 <2226142553 bps
	omd site demo	OK	14:09:42	1m 19s	1/3	OK - site demo: running
	os version	OK	14:09:04	57s	1/3	OK: debian 12.1 (arch: arm64)
	service apache2	OK	14:09:03	58s	1/3	OK: All 1 service(s) are ok.
	service mariadb	OK	14:09:13	1m 48s	1/3	OK: All 1 service(s) are ok.
	service postfix	OK	14:09:56	1m 5s	1/3	OK: All 1 service(s) are ok.
	service snclient	OK	14:09:27	1m 34s	1/3	OK: All 1 service(s) are ok.
	service ssh	OK	14:09:03	58s	1/3	OK: All 1 service(s) are ok.
	uptime	WARNING	14:08:27	1m 34s	1/3	WARNING: uptime: 1d 05:30h, boot: 2023-11-06 07:37:5...

14 of 14 Items Displayed

Installation

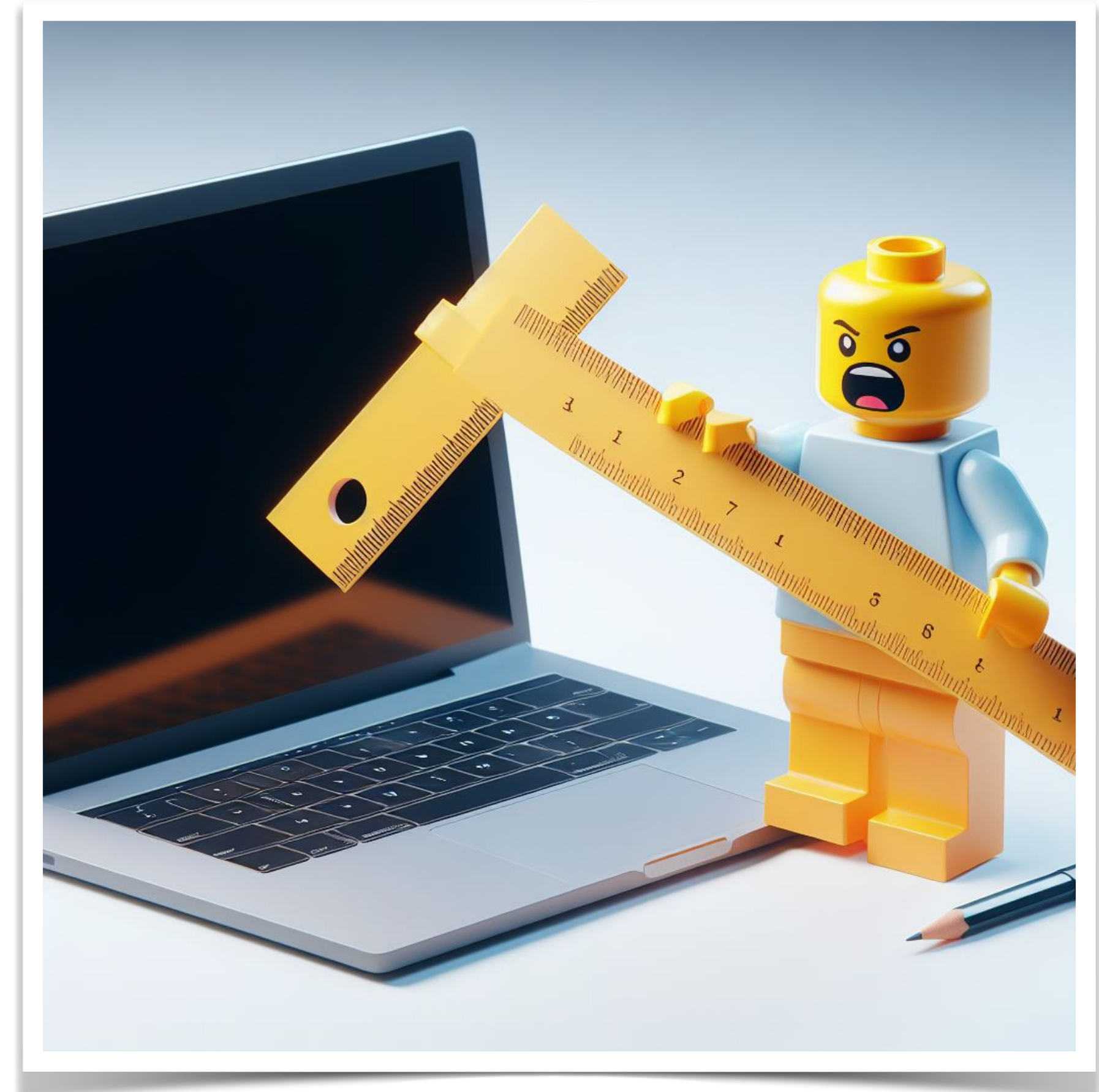
- .MSI Installer for windows
- DEB/RPM repository for linux:
<https://labs.consol.de/repo/>
- Single go binary

```
ssh
10:46:22 root@legerarmvm:~ 9 #>apt install snclient
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  snclient
0 upgraded, 1 newly installed, 0 to remove and 96 not upgrade
Need to get 11.8 MB of archives.
After this operation, 0 B of additional disk space will be us
Get:1 http://labs.consol.de/repo/testing/debian bookworm/main
Fetched 11.8 MB in 0s (72.9 MB/s)
Selecting previously unselected package snclient.
```



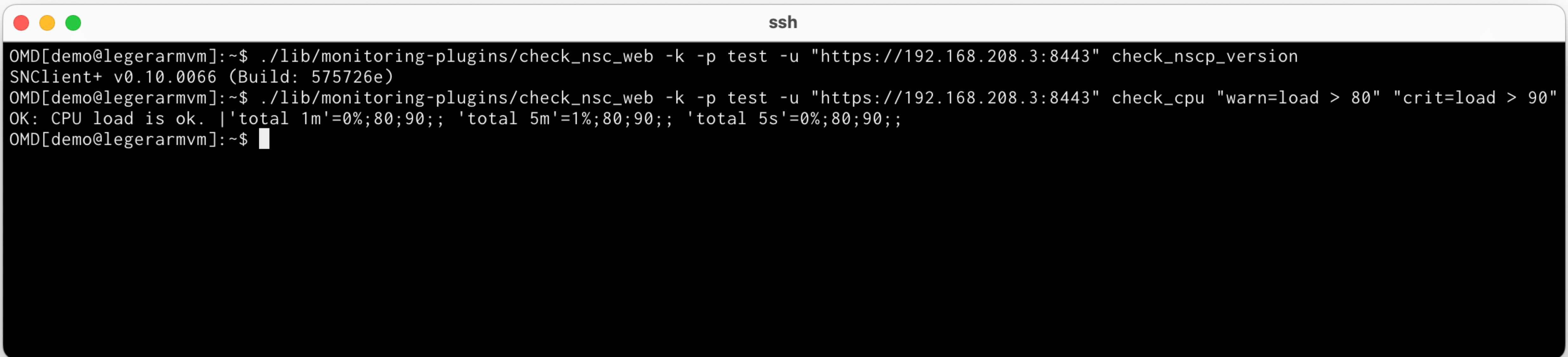
Checks

- Built in check commands, ex.:
 - `check_cpu`
 - `check_drivesize`
 - `check_memory`
 - `check_process`
 - `check_service`
 - ...
- Get full list with
 - `.\snclient.exe run check_index`



Checks

- Checks can be executed via `check_nsc_web` utilising the REST API.

A terminal window titled 'ssh' with standard macOS window controls (red, yellow, green buttons). The terminal shows a user at 'demo@legerarmvm' running two commands. The first command is './lib/monitoring-plugins/check_nsc_web -k -p test -u "https://192.168.208.3:8443" check_nscp_version', which outputs 'SNClient+ v0.10.0066 (Build: 575726e)'. The second command is './lib/monitoring-plugins/check_nsc_web -k -p test -u "https://192.168.208.3:8443" check_cpu "warn=load > 80" "crit=load > 90"', which outputs 'OK: CPU load is ok. |'total 1m'=0%;80;90;; 'total 5m'=1%;80;90;; 'total 5s'=0%;80;90;;'.

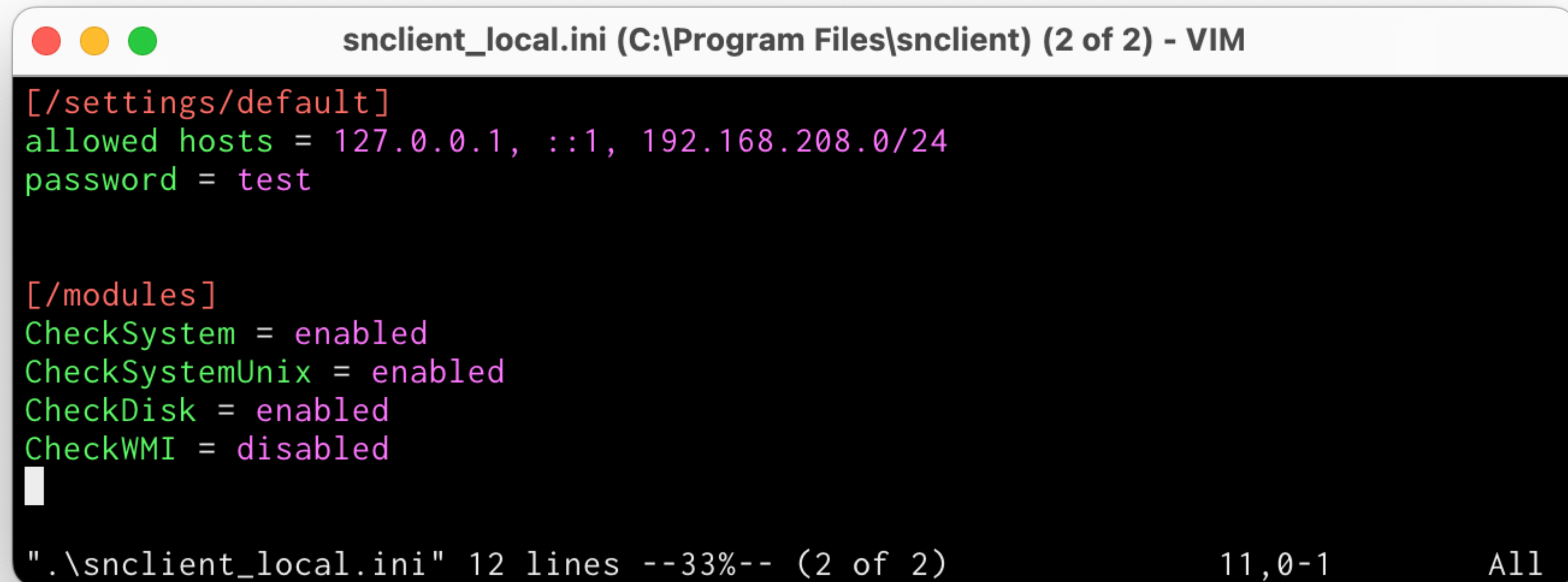
```
OMD[demo@legerarmvm]:~$ ./lib/monitoring-plugins/check_nsc_web -k -p test -u "https://192.168.208.3:8443" check_nscp_version
SNClient+ v0.10.0066 (Build: 575726e)
OMD[demo@legerarmvm]:~$ ./lib/monitoring-plugins/check_nsc_web -k -p test -u "https://192.168.208.3:8443" check_cpu "warn=load > 80" "crit=load > 90"
OK: CPU load is ok. |'total 1m'=0%;80;90;; 'total 5m'=1%;80;90;; 'total 5s'=0%;80;90;;
OMD[demo@legerarmvm]:~$
```

- NRPE is available for compatibility reasons, `check_nsc_web` is the recommended way.

https://github.com/ConSol-Monitoring/check_nsc_web

Checks

- Some checks need to be enabled in the ["/modules]" section.



```
snclient_local.ini (C:\Program Files\snclient) (2 of 2) - VIM

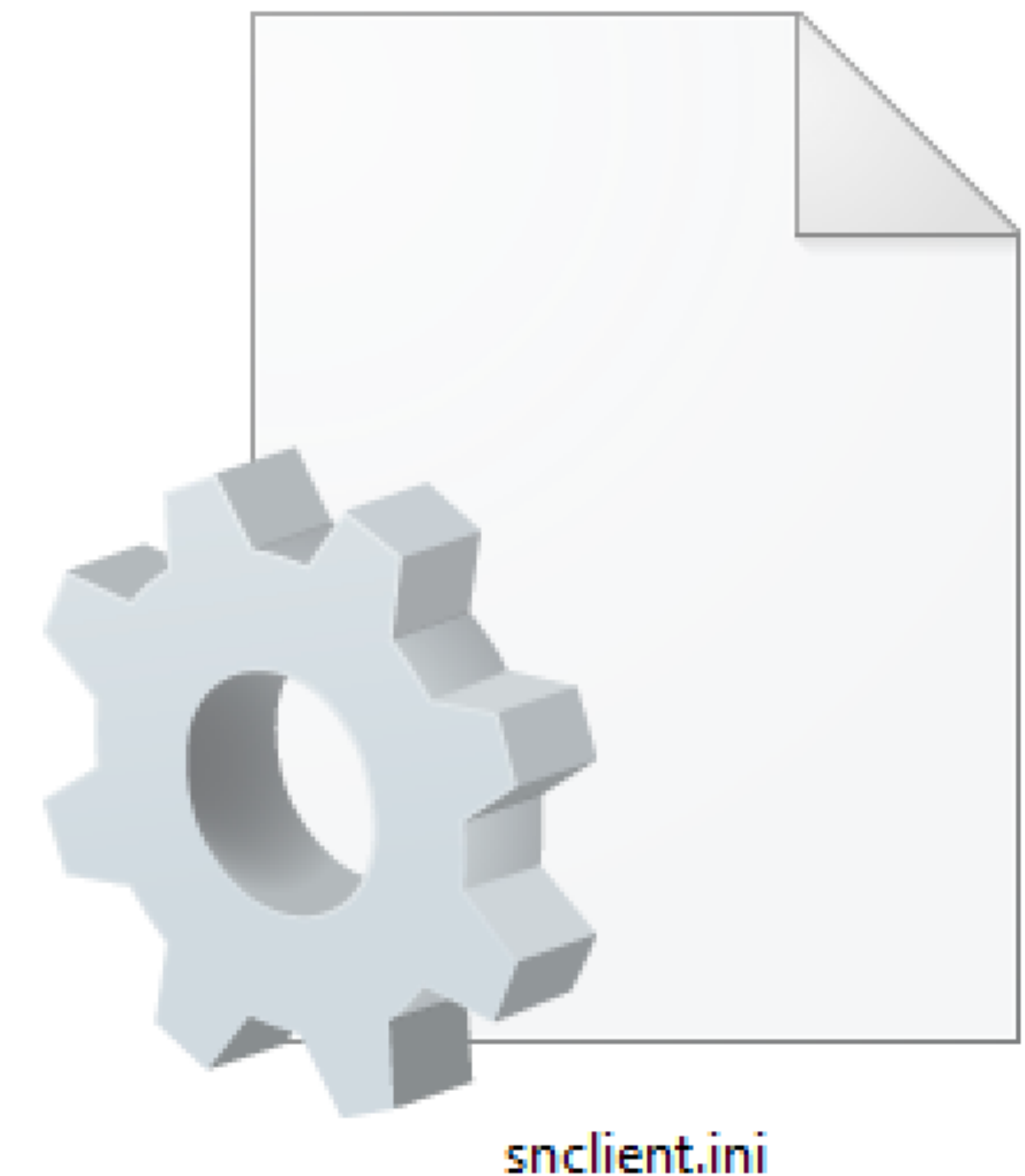
[/settings/default]
allowed hosts = 127.0.0.1, ::1, 192.168.208.0/24
password = test

[/modules]
CheckSystem = enabled
CheckSystemUnix = enabled
CheckDisk = enabled
CheckWMI = disabled

".\snclient_local.ini" 12 lines --33%-- (2 of 2) 11,0-1 All
```

Configuration

- Default snclient.ini contains all available values with their defaults
- Best practice: Put changes into a snclient_local.ini which is included by default
- Included files overwrite existing settings



Configuration - Inheritance

- The order of inheritance for the example to the right:
 - /settings/sub1/other (most significant)
 - /settings/sub1/default
 - /settings/sub1
 - /settings/default (least significant)

```
[/settings/sub1/other]  
key = value
```

```
[/settings/sub1/default]  
; fallback if the above is not set  
key = value
```

```
[/settings/sub1]  
; fallback if the above is not set  
key = value
```

```
[/settings/default]  
; fallback if the above is not set  
key = value
```

Configuration - Macros

- Default macros:
 - `${exe-path}`
 - `${shared-path}`
 - `${scripts}`
 - `${certificate-path}`
 - `${hostname}`
 - `${goos}` and `${goarch}`
 - `%(macroname)` is also supported
- Example:
[`/settings/log`]
file name = `${shared-path}/snclient.log`



Configuration - On Demand Macros

- On demand macros can be used to access values from other sections
- Example:

```
[/settings/external scripts/alias]  
alias_allowed_hosts = check_dummy 0 "${/settings/WEB/server/allowed hosts}"
```
- Those macros are only valid in the .ini file, not as check arguments.

Configuration - Macros Operators

- Use macro operators to alter existing values:
- Example:
[/settings/external scripts/alias]
alias_hostname = check_dummy 0 "host:\${hostname:lc}"

<https://omd.consol.de/docs/snclient/configuration/#macro-operators>

Checks - Filter

- Each check builds an internal table for detail data:
.\snclient.exe run -vvv check_service filter="name=snclient"
OK: All 1 service(s) are ok.

classification	cpu	delayed	desc	name	pid	service	start_type	state
service-shared-process	0 %	0	AssignedAccessManager	AssignedAccessManagerSvc	0	AssignedAccessManagerSvc	demand	stopped
service-own-process	1 %	0	Windows Audio	Audiosrv	1700	Audiosrv	auto	running
service-own-process	0.2%	0	snclient	snclient	8336	snclient	Auto	Running
...								

- Filter are simple "<key> <op> <value>" sets combined by logical operators

<https://omd.consol.de/docs/snclient/checks/filter/>

Checks - Thresholds

- Thresholds work like filter but instead of removing an entry, the according state will be set.

```
.\snclient.exe run check_service filter="name=snclient" "warn=cpu>90%" "crit=cpu>95" show-all
```

```
OK: snclient=running (auto) |'snclient'=4 'snclient rss'=32395264B 'snclient vms'=29831168B 'snclient cpu'=0.1%
```

- Use multiple warn/crit to create ranges

```
.\snclient.exe run check_drivesize drive=c: warn="used > 10% && used < 30%"
```

```
OK: All 1 drive(s) are ok |... 'c: used %'=57.9%;@10:30;90;0;100
```



External Scripts

- External scripts need to be enabled in the “[/modules]” section first:

```
[/modules]
```

```
CheckExternalScripts = enabled
```

- Concise Format:

```
[/settings/external scripts]
```

```
my_check1 = check_custom.bat
```

```
my_check2 = myscripts\check_custom.bat
```

- Verbose Format:

```
[/settings/external scripts/scripts/my_check1]
```

```
my_check1 = check_custom.bat
```

```
[/settings/external scripts/scripts/my_check2]
```

```
my_check2 = myscripts\check_custom.bat
```



External Scripts

- Don't:

```
cmd /c echo scripts\check_something.ps1 $ARG1$; exit($lastexitcode) | powershell.exe -command -
```

- Instead:

```
scripts\check_something.ps1 $ARG1$
```



Plugins

- Plugins are built in checks, but work more like the official monitoring-plugins
- Currently there are:
 - `check_nsc_web`
 - `check_http`
 - `check_tcp`
 - `check_dns`
- Plugins need to be enabled:
[`/modules`]
`CheckBuiltinPlugins = enabled`



<https://github.com/mackerelio/go-check-plugins>

Plugins

- Example: use internal `check_http` to check reachability of services:

```
.\snclient.exe run check_http -H company-service:8443 -S
```

```
HTTP OK: HTTP/1.1 307 Temporary Redirect - 202 bytes in 0.073 second response time | time=0.072689s...
```

```
.../check_nsc_web -p 123 -u https://hostname:8443 check_http -H labs.consol.de -S
```

- Example: use internal `check_nsc_web` to check other snclients:

```
.\snclient.exe run check_nsc_web -p test -u https://192.123.123.2:8443 -k check_os_version
```

```
OK: Microsoft Windows 11 Pro 10.0.22631.2428 Build 22631.2428 (arch: amd64)
```

Security

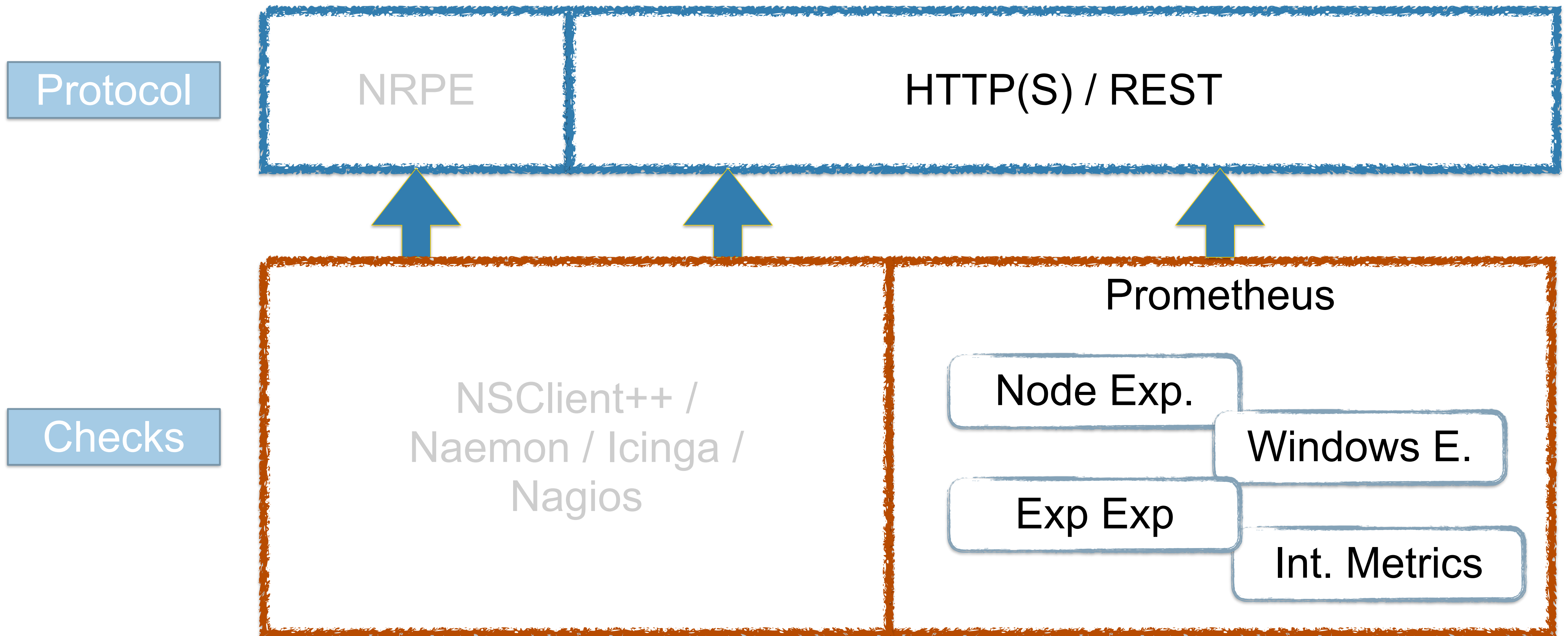
- Native SSL/TLS Support
- Strong Encryption and Security Standards
- Secure Ciphers by Default
- Hashed Passwords:
[`/settings/default`]
`password = SHA256:9f86d081...`
- “Allowed Hosts” are used for the webserver as well (all listeners)





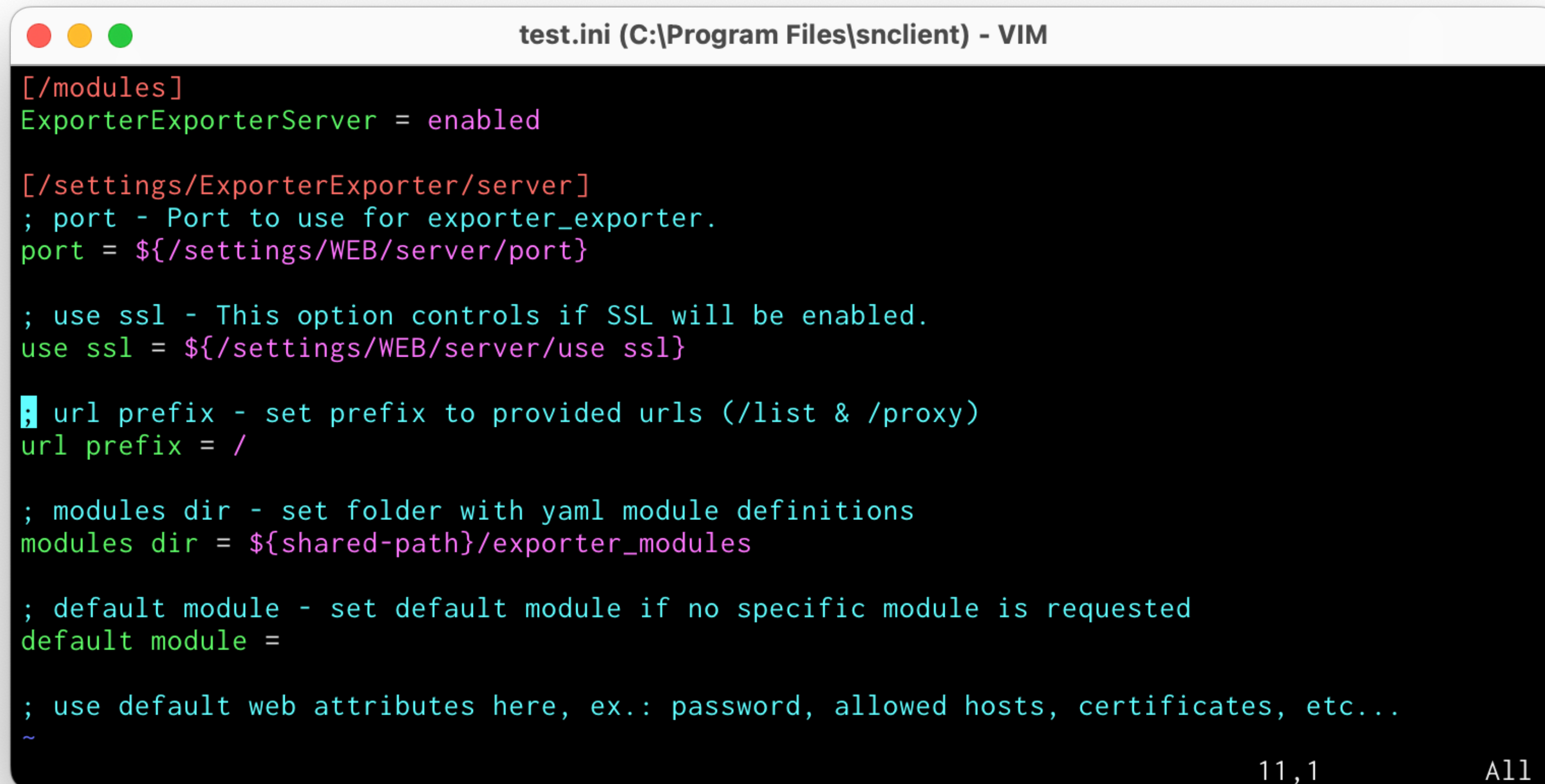
Prometheus Stack

Prometheus Stack



Prometheus Stack

- Exporter Exporter
 - Basically a reverse proxy for prometheus exporters



```
test.ini (C:\Program Files\snclient) - VIM

[/modules]
ExporterExporterServer = enabled

[/settings/ExporterExporter/server]
; port - Port to use for exporter_exporter.
port = ${/settings/WEB/server/port}

; use ssl - This option controls if SSL will be enabled.
use ssl = ${/settings/WEB/server/use ssl}

; url prefix - set prefix to provided urls (/list & /proxy)
url prefix = /

; modules dir - set folder with yaml module definitions
modules dir = ${shared-path}/exporter_modules

; default module - set default module if no specific module is requested
default module =

; use default web attributes here, ex.: password, allowed hosts, certificates, etc...
~

11,1 All
```

Prometheus Stack

- Exporter Exporter
 - Uses .yaml format for compatibility reasons in the “modules dir”

- Example `.\exporter_modules\client.yaml`:

method: http

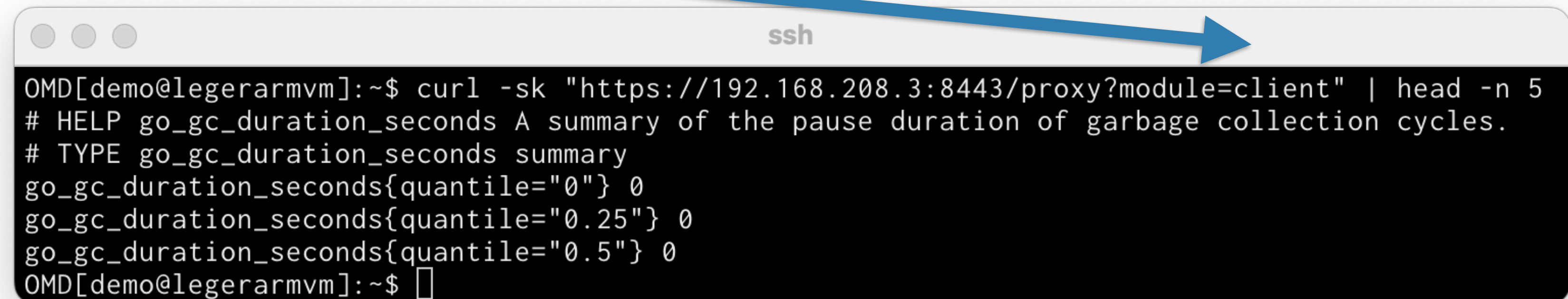
http:

 scheme: http

 port: 9999

 path: /metrics

 tls_insecure_skip_verify: true



A terminal window titled 'ssh' showing a command and its output. A blue arrow points from the text 'client.yaml' in the list above to the terminal window. The terminal output shows the first five lines of a Prometheus metrics response for the 'go_gc_duration_seconds' metric.

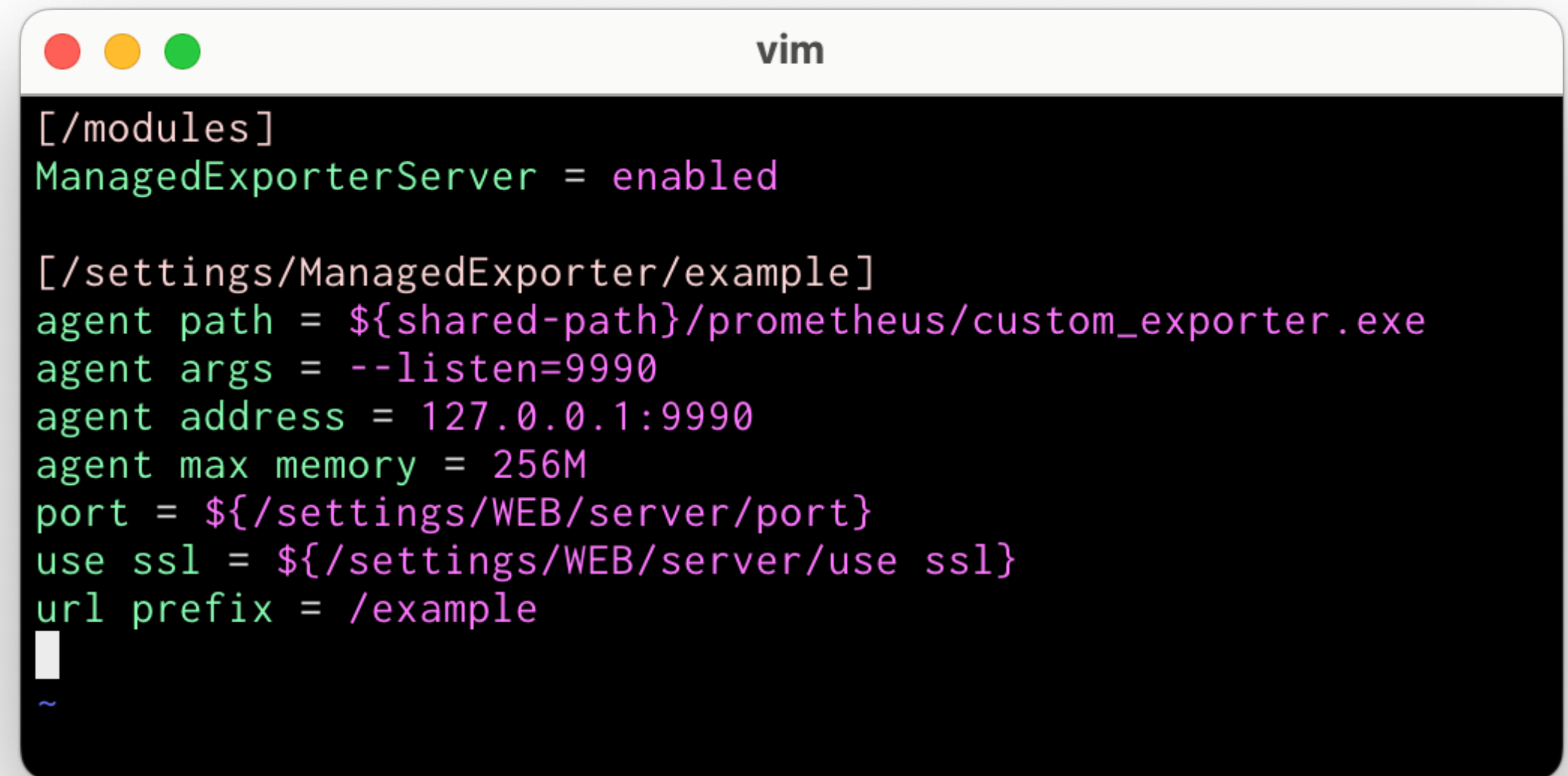
```
OMD[demo@legerarmvm]:~$ curl -sk "https://192.168.208.3:8443/proxy?module=client" | head -n 5
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
OMD[demo@legerarmvm]:~$
```

- Does not verify metrics and just passes through

https://github.com/QubitProducts/exporter_exporter

Prometheus Stack

- Managed Exporter
 - Starts and controls exporters
 - Controls memory usage
 - Restarts exporter in case of errors
 - Enable with
“ManagedExporterServer = enabled”
in the modules section
 - URL Prefix /example usually results in
/example/metrics as path to metrics

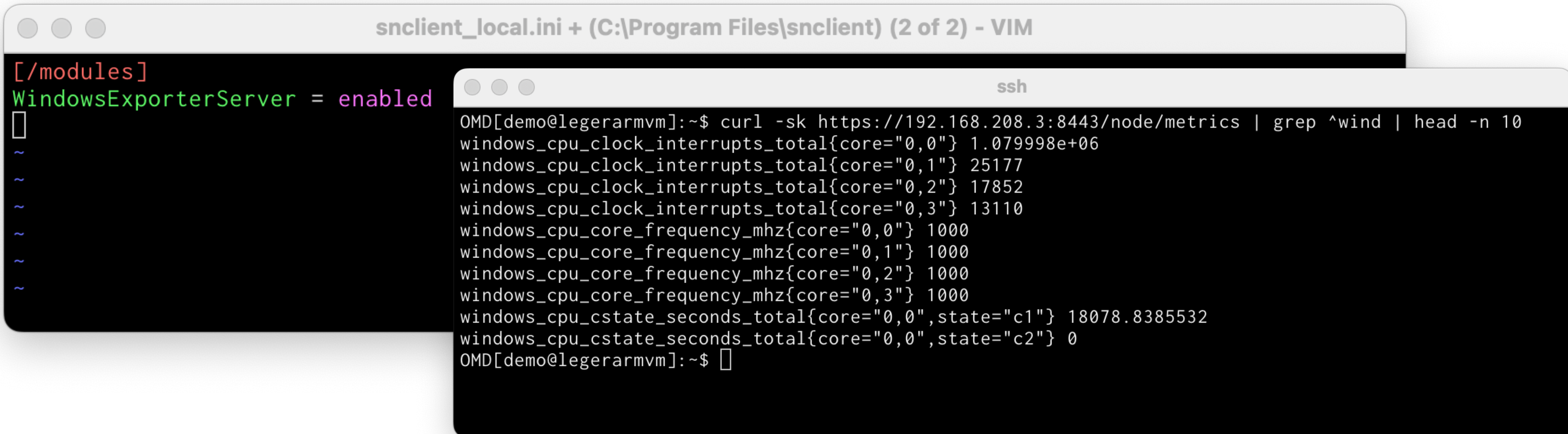


```
vim
[/modules]
ManagedExporterServer = enabled

[/settings/ManagedExporter/example]
agent path = ${shared-path}/prometheus/custom_exporter.exe
agent args = --listen=9990
agent address = 127.0.0.1:9990
agent max memory = 256M
port = ${/settings/WEB/server/port}
use ssl = ${/settings/WEB/server/use ssl}
url prefix = /example
~
```

Prometheus Stack

- Windows Exporter
 - Managed Exporter
 - Started and controlled by SNClient+
 - Checks memory usage and restarts the exporter

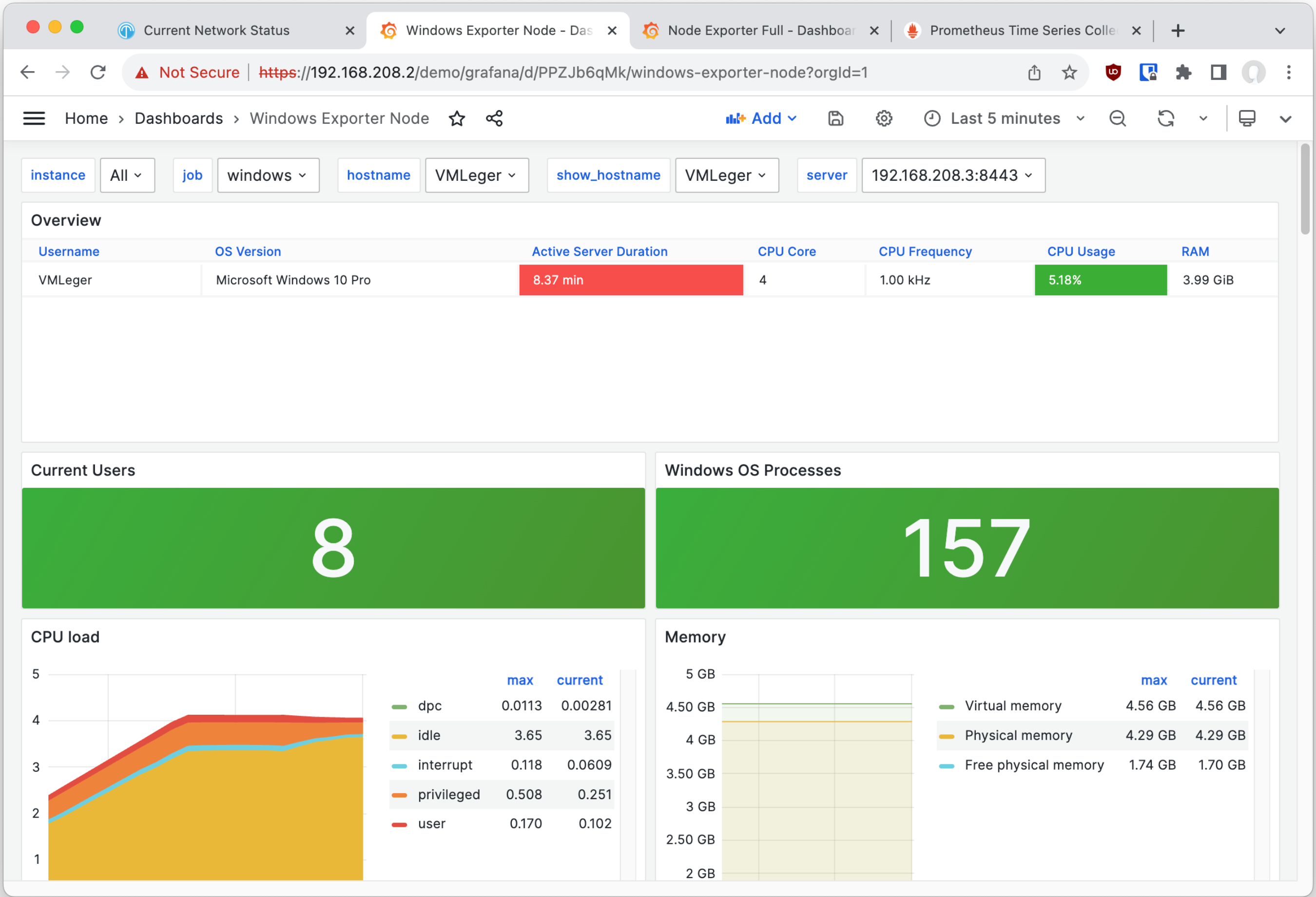


The image shows two overlapping terminal windows. The background window is a VIM editor editing 'snclient_local.ini'. The foreground window is an SSH terminal running a curl command to fetch Prometheus metrics from a Windows exporter.

```
snclient_local.ini + (C:\Program Files\snclient) (2 of 2) - VIM
[/modules]
WindowsExporterServer = enabled
~
~
~
~
~
~

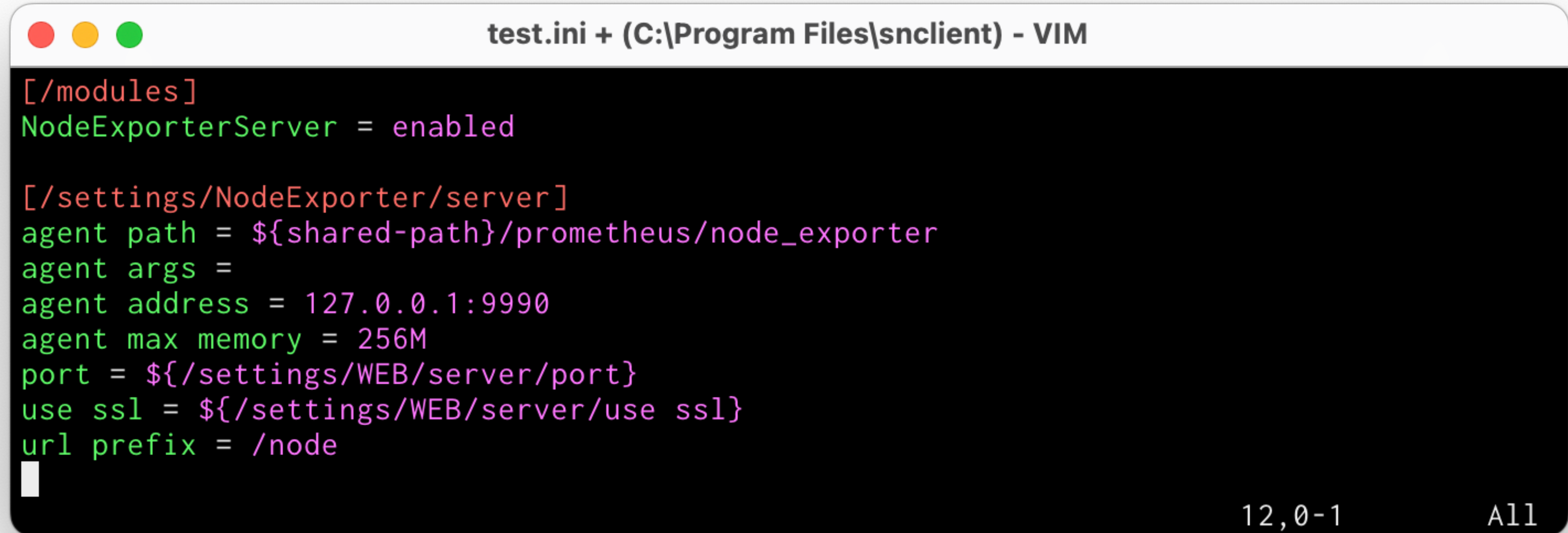
ssh
OMD[demo@legerarmvm]:~$ curl -sk https://192.168.208.3:8443/node/metrics | grep ^wind | head -n 10
windows_cpu_clock_interrupts_total{core="0,0"} 1.079998e+06
windows_cpu_clock_interrupts_total{core="0,1"} 25177
windows_cpu_clock_interrupts_total{core="0,2"} 17852
windows_cpu_clock_interrupts_total{core="0,3"} 13110
windows_cpu_core_frequency_mhz{core="0,0"} 1000
windows_cpu_core_frequency_mhz{core="0,1"} 1000
windows_cpu_core_frequency_mhz{core="0,2"} 1000
windows_cpu_core_frequency_mhz{core="0,3"} 1000
windows_cpu_cstate_seconds_total{core="0,0",state="c1"} 18078.8385532
windows_cpu_cstate_seconds_total{core="0,0",state="c2"} 0
OMD[demo@legerarmvm]:~$
```

Prometheus Stack



Prometheus Stack

- Node Exporter
 - Same as windows exporter but for linux



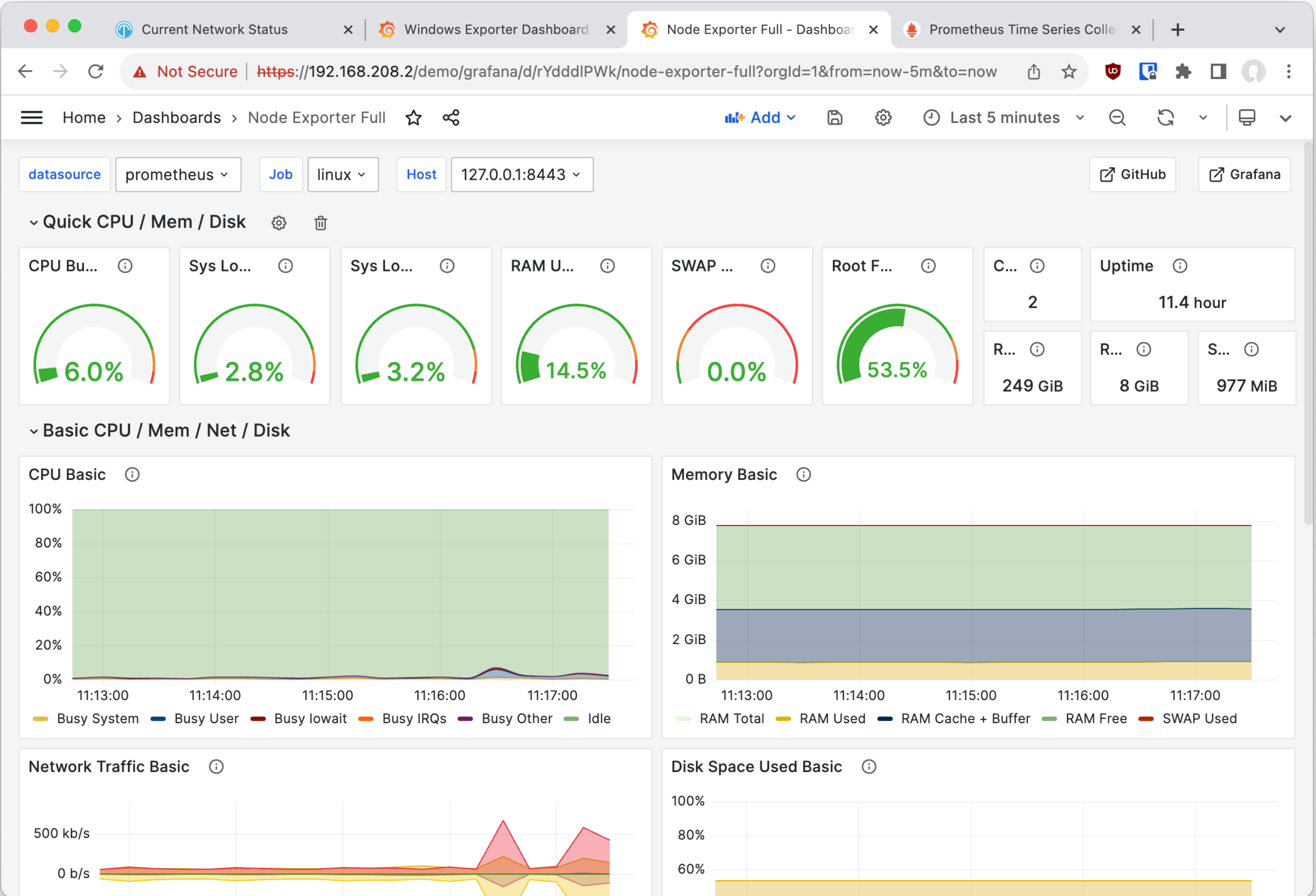
The screenshot shows a VIM editor window titled "test.ini + (C:\Program Files\snclient) - VIM". The editor displays the following configuration for Node Exporter:

```
[/modules]
NodeExporterServer = enabled

[/settings/NodeExporter/server]
agent path = ${shared-path}/prometheus/node_exporter
agent args =
agent address = 127.0.0.1:9990
agent max memory = 256M
port = ${/settings/WEB/server/port}
use ssl = ${/settings/WEB/server/use ssl}
url prefix = /node
```

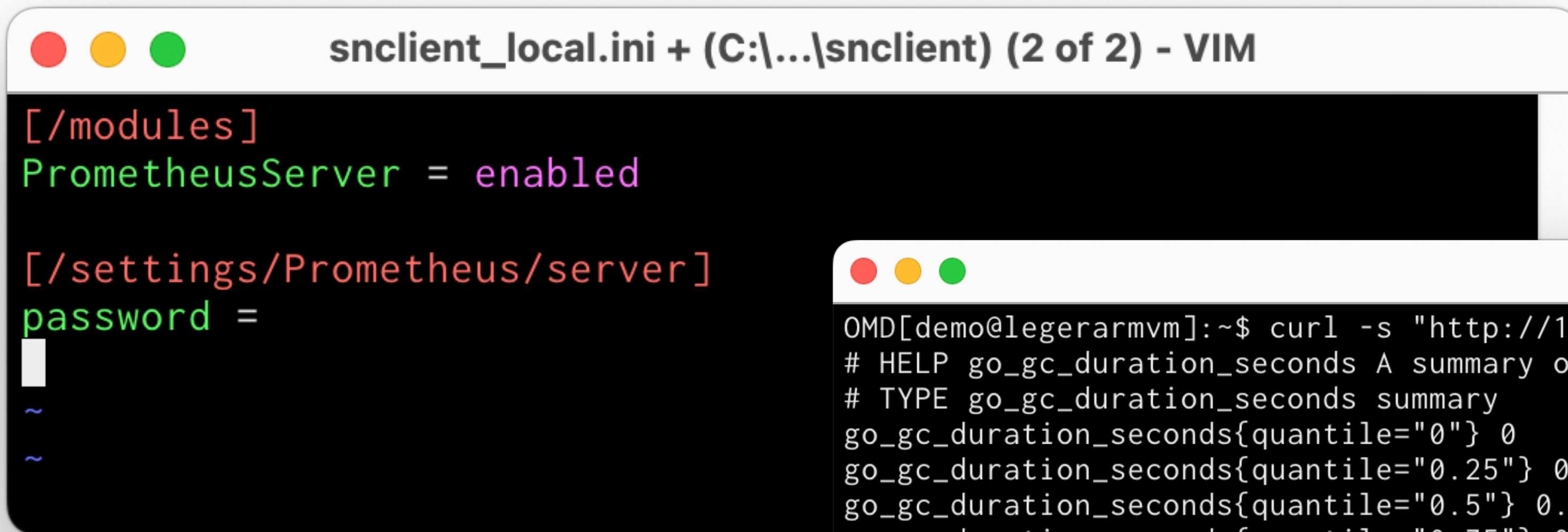
The status bar at the bottom right of the VIM window shows "12,0-1" and "All".

Prometheus Stack



Prometheus Stack

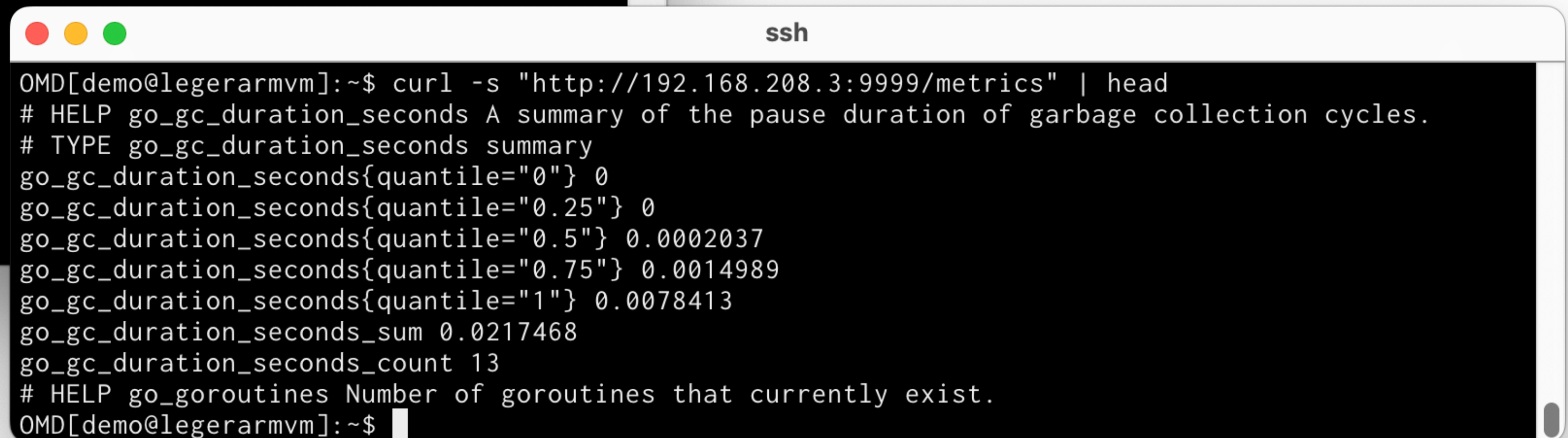
- Like every good tool in the prometheus eco system
 - Provides internal metrics
 - Must be enabled with “PrometheusServer” in the modules section.



```
snclient_local.ini + (C:\...\snclient) (2 of 2) - VIM

[/modules]
PrometheusServer = enabled

[/settings/Prometheus/server]
password =
~
~
```



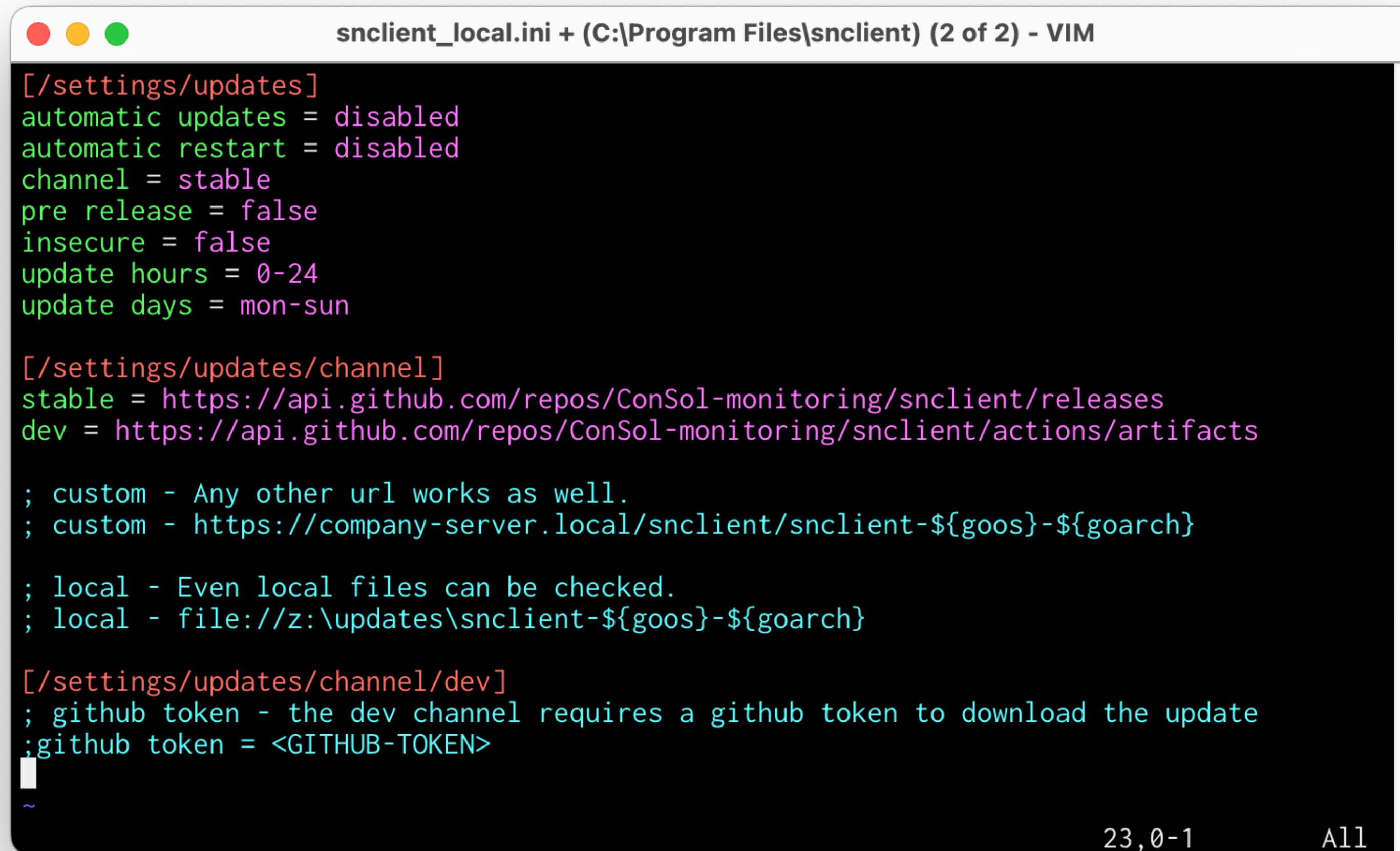
```
ssh

OMD[demo@legerarmvm]:~$ curl -s "http://192.168.208.3:9999/metrics" | head
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0.0002037
go_gc_duration_seconds{quantile="0.75"} 0.0014989
go_gc_duration_seconds{quantile="1"} 0.0078413
go_gc_duration_seconds_sum 0.0217468
go_gc_duration_seconds_count 13
# HELP go_goroutines Number of goroutines that currently exist.
OMD[demo@legerarmvm]:~$
```



Updates

Updates



```
snclient_local.ini + (C:\Program Files\snclient) (2 of 2) - VIM

[/settings/updates]
automatic updates = disabled
automatic restart = disabled
channel = stable
pre release = false
insecure = false
update hours = 0-24
update days = mon-sun

[/settings/updates/channel]
stable = https://api.github.com/repos/ConSol-monitoring/snclient/releases
dev = https://api.github.com/repos/ConSol-monitoring/snclient/actions/artifacts

; custom - Any other url works as well.
; custom - https://company-server.local/snclient/snclient-${goos}-${goarch}

; local - Even local files can be checked.
; local - file://z:\updates\snclient-${goos}-${goarch}

[/settings/updates/channel/dev]
; github token - the dev channel requires a github token to download the update
;github token = <GITHUB-TOKEN>

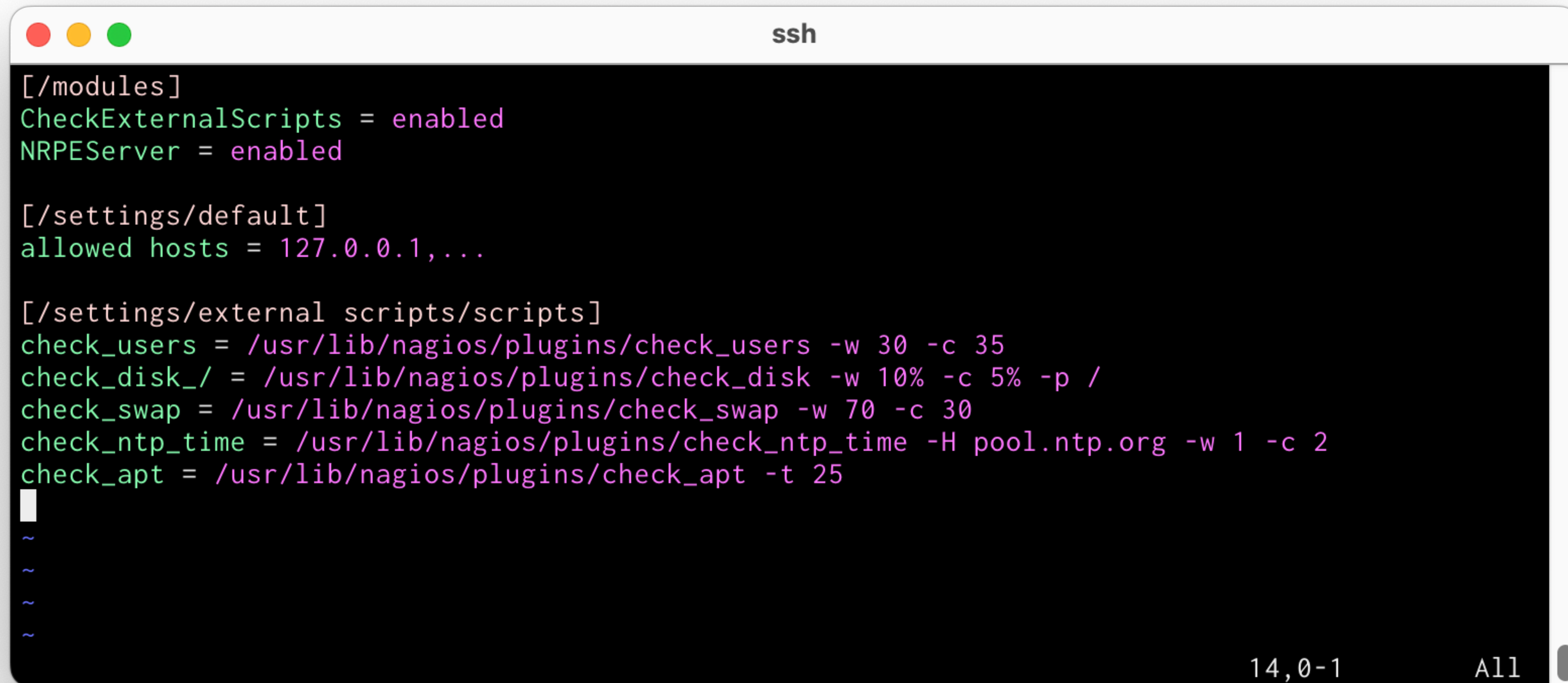
~

23,0-1 All
```



Examples

Replace NRPE



```
[~/modules]
CheckExternalScripts = enabled
NRPEServer = enabled

[/settings/default]
allowed hosts = 127.0.0.1,...

[/settings/external scripts/scripts]
check_users = /usr/lib/nagios/plugins/check_users -w 30 -c 35
check_disk_/ = /usr/lib/nagios/plugins/check_disk -w 10% -c 5% -p /
check_swap = /usr/lib/nagios/plugins/check_swap -w 70 -c 30
check_ntp_time = /usr/lib/nagios/plugins/check_ntp_time -H pool.ntp.org -w 1 -c 2
check_apt = /usr/lib/nagios/plugins/check_apt -t 25
~
~
~
~
```

14,0-1 All



Performance

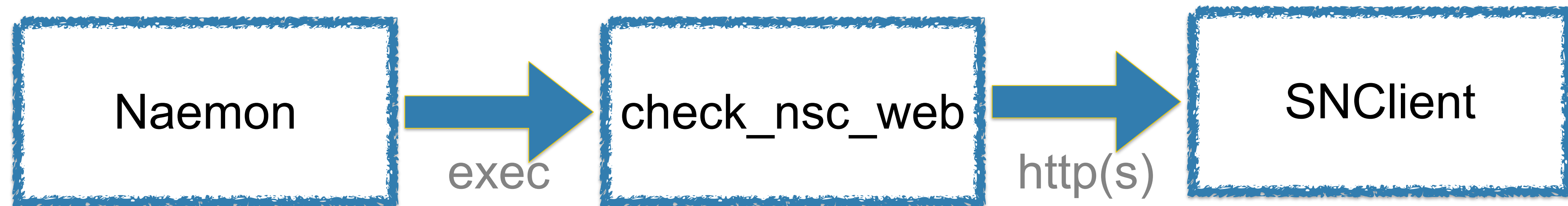
Performance

- Best practice:
 - Naemon Core
 - check_nsc_web
 - Mod-Gearman (Go) Worker
- But everything else is fine as well



Performance

Classical Approach:



Mod-Gearman:



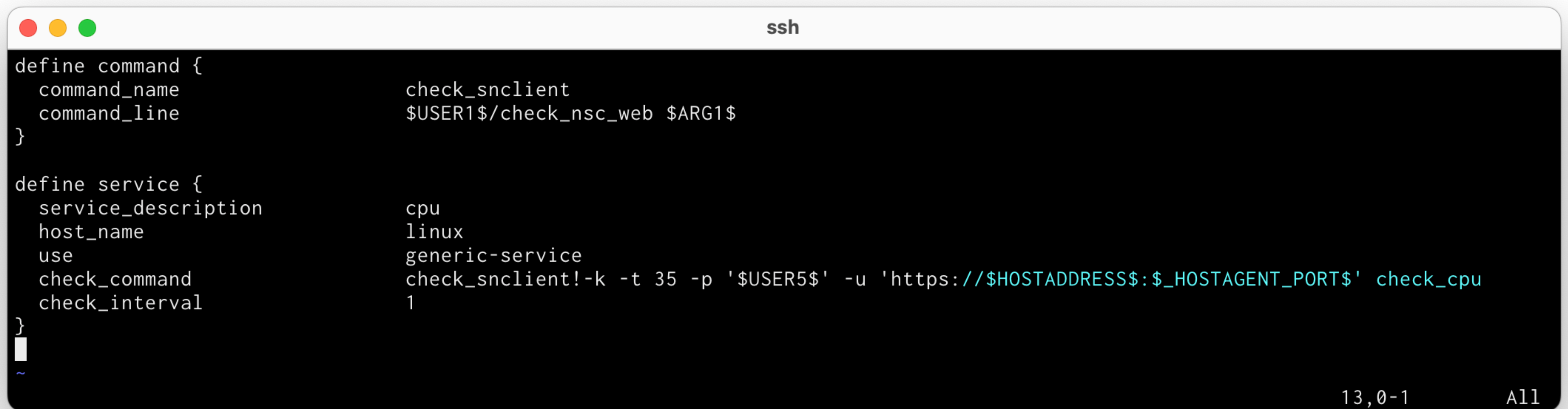
The Mod-Gearman Go Worker uses a builtin check_nsc_web



Thruk Agents

Thruk Agents

- This is fine, but...

A terminal window titled 'ssh' with a dark background and light green text. It displays a configuration script for a Thruk agent. The script defines a command 'check_snclient' and a service 'cpu'. The service configuration includes host_name 'linux', use 'generic-service', check_command 'check_snclient!-k -t 35 -p '\$USER5\$' -u 'https://\$HOSTADDRESS\$:\$_HOSTAGENT_PORT\$' check_cpu', and check_interval '1'. The terminal window has standard macOS window controls (red, yellow, green buttons) in the top left corner. At the bottom right, it shows '13,0-1' and 'All'.

```
ssh

define command {
    command_name      check_snclient
    command_line      $USER1$/check_nsc_web $ARG1$
}

define service {
    service_description    cpu
    host_name              linux
    use                    generic-service
    check_command          check_snclient!-k -t 35 -p '$USER5$' -u 'https://$HOSTADDRESS$:$_HOSTAGENT_PORT$' check_cpu
    check_interval         1
}

~
```

13,0-1 All

Thruk Agents

The screenshot shows the Thruk Agents web interface in a browser. The browser's address bar displays the URL `https://192.168.208.2/demo/thruk/cgi-bin/agents.cgi`. The interface features a sidebar on the left with a navigation menu containing sections like General, Current Status, Reports, and System. The 'Agents' section is currently selected. The main content area displays a table of agents with columns for Section, Hostname, Status, Site, Agent, and Version. Two agents are listed: one for 'linux' and one for 'windows', both with a status of 'OK'. A green button labeled '+ add new host' is located in the top right corner of the table area. The browser's tab bar shows several open tabs, including 'Agents', 'Windows Exporter Node - Das', 'Node Exporter Full - Dashboar', and 'Prometheus Time Series Collec'.

filter

Agents

+ add new host

Section	Hostname	Status	Site	Agent	Version	
	linux	OK	demo	snclient	v0.11.0023 (Build: cc5501f)	 
	windows	OK	demo	snclient	v0.10.0066 (Build: 575726e)	 

2 of 2 Items Displayed

Thruk Agents

Agents

Windows Exporter Node - Das

Node Exporter Full - Dashboar

Prometheus Time Series Collec

Not Secure | https://192.168.208.2/demo/thruk/cgi-bin/agents.cgi?action=edit&hostname=linux&backend=48f1c

search...

refresh

settings

external-link

thruk

General

Current Status

Tactical Overview

Hosts

Services

Host Groups

Service Groups

Mine Map

Problems

Reports

System

Comments

Downtimes

Process Info

Performance Info

Scheduling Queue

Configuration

Broadcasts

Agents

Config Tool

Agents

« Back

Edit Agent

Activate Changes

Hostnamelinux

Section:

IP Address127.0.0.1

Port:8443

Passwordpassword set, enter new one to change

Discover Services

Save Changes

Add New (0)

Active Checks (14)

Obsolete Checks (0)

Unwanted Checks (32)

agent version	OK		SNClient+ v0.11.0023 (Build: cc5501f)	On	Off
cpu	OK		OK: CPU load is ok.	On	Off
memory	OK		OK: committed = 0 B, physical = 939.01 MiB	On	Off
uptime	CRITICAL		CRITICAL: uptime: 14:04h, boot: 2023-11-07 22:50:47 (UTC)	On	Off
os version	OK		OK: debian 12.1 (arch: arm64)	On	Off
net enp0s1	OK		(info) OK: enp0s1 >40517685 <254150204 bps	On	Off
disk /	OK		(info) OK: All 1 drive(s) are ok	On	Off
service apache2	OK		(info) OK: All 1 service(s) are ok.	On	Off
service mariadb	OK		(info) OK: All 1 service(s) are ok.	On	Off
service postfix	OK		(info) OK: All 1 service(s) are ok.	On	Off
service snclient	OK		(info) OK: All 1 service(s) are ok.	On	Off

Delete



Roadmap

Roadmap

- Improve defaults for builtin checks
- Add more builtin checks
- Add configuration validator
- Maybe add support for opentelemetry



Thanks



- <https://github.com/ConSol-Monitoring/snclient>
- <https://omd.consol.de/docs/snclient/>
- <mailto:sven.nierlein@consol.de>

