Eulenfunk Documentation

Release 0.1

JJX, Adorfer, Benedikt Wildenhain, tuennes, PetaByteBoy

July 02, 2016

Contents

1 Communities **3**
- 1.1 Organigramm . . . 3
- 1.2 Fichtenfunk . . . 4
- 1.3 Siegerland . . . 6
- 1.4 Neanderfunk . . . 7
- 1.5 Düsseldorf-Flingern . . . 9
- 1.6 GL.Eulenfunk . . . 10

2 Infrastruktur **11**
- 2.1 gemietete Hosts . . . 11
- 2.2 Virtuelle Hosts . . . 11

3 Public Keys (people) **13**
- 3.1 ssh . . . 13
- 3.2 firmware . . . 15

4 Firmware **17**
- 4.1 Flingern . . . 17

5 Server Kochbuch **19**
- 5.1 Das Blech . . . 19
- 5.2 Proxmox . . . 20
- 5.3 BGP Konzentrator einrichten . . . 25
- 5.4 Supernode einrichten . . . 33
- 5.5 Nützliches . . . 41

6 Map **43**

Was ist Eulenfunk nicht?

- keine Community
- keine schlüsselfertige Lösung
- es gibt nichts umsonst

Was leistet Eulenfunk nicht?

- Eulenfunk administriert keine Server für euch
- Eulenfunk ist kein 24/7 Support-Dienstleister
- Eulenfunk hilft euch nicht beim Kleinreden machbarer Projekte
- Eulenfunk brät dir nicht deine Extrawurst
- Eulenfunk stellt keine Router in deiner Stadt auf

Wer sollte hier nicht mitmachen?

Du wirst beim eulenfunk vermutlich nicht glücklich wenn Du

- Deine Arbeit nicht öffentlich und zeitnah dokumentierst
- nicht kritikfähig bist
- wenn Du schon alles ganz genau weisst
- nicht regelmäßig im Mumble bist
- nicht nachtaktiv bist
- keine Zeit in das Teilen von Wissen investieren magst
- kein Vertrauen wagst in die Fähigkeiten von anderen
- Tooldiskussionen liebst
- lieber Lösungswege diskutierst als sie praktisch evaluierst.
- die Welt erklären kannst, insbesondere wenn alle anderen böse sind.
- es Dir wichtig ist, besser als eine andere Person oder Gruppe zu sein.

Willst Du immernoch? (Ja / Nein / Vielleicht)

Wenn Du also bereit bis, Dir mit uns gemeinsam in totaler Frustresistenz die Nächte um die Ohren zu schlagen.

- Und am nächsten Tag dann hören darfst “Alles(tm) funktioniert nicht”.
- Und dann eine informative, freundliche Erklärung präsentieren kannst.
- Und wenn Du zum gemeinsamen Kochen auch noch was mitbringst,

Dann komm doch mal ins allabendliche Eulenmumble: mumble.eulenfunk.de:64738 (mumble vom Freifunk Rheinland)

```
^_^
{o,o}
|)__)
-m-m-
```

CHAPTER 1

Communities

1.1 Organigramm

- Flingern
- **Fichtenfunk**
  - Altena (al)
  - Halver (hv)
  - Hemer (he)
  - Herscheid (he)
  - Iserlohn (is)
  - Lüdenscheid (ls)
  - Meinerzhagen (mz)
  - Menden (md)
  - Neuenrade
  - Plettenberg
  - Schalksmühle (sm)
  - Werdohl
  - Bergneustadt
  - Gummersbach
  - Marienheide
- **GL.Eulenfunk**
  - Leichlingen (lln)
  - Bergisch-Gladbach (bgl)
- **Neanderfunk**
  - Erkrath (erk)
  - Haan (han)
  - Hilden (hld)

  - Heiligenhaus (hlg)
  - Langenfeld (lgf)
  - Mettmann (met)
  - Monheim (mon)
  - Ratingen (rat)
  - Velbert (vel)
  - Wülfrath (wlf)
- **Siegerland**
  - Bad Berleburg
  - Burbach
  - Erntebrück
  - Hilchenbach (hb)
  - Kreuztal
  - Netphen
  - Siegen (si)
  - Wilnsdorf

1.2 Fichtenfunk

Fichtenfunk ist Freifunk im Märkischen Kreis von Iserlohn bis Meinerzhagen.

AS	65410
IPv6	2a03:2260:120::/48

1.2.1 Websites

Karte	http://map.freifunk-mk.de
Firmware	http://firmware.freifunk-mk.de
Webseiten	http://freifunk-mk.de
	http://freifunk-iserlohn.de
	http://freifunk-hemer.de
	http://freifunk-altena.de
	http://freifunk-meinerzhagen.de

1.2.2 Team

Nick	Name	Kontakt	Was
domfi			
ling			
jjx		julian.bog@freifunk-mk.de	

1.2.3 Bleche

Community/Blech	1	2	3	4
Altena		x		
Halver				
Hemer	x			
Herscheid				
Iserlohn	x			
Lüdenscheid	x			
Meinerzhagen		x		
Menden				
Neuenrade				
Plettenberg				
Schalksmühle				
Werdohl				

1.2.4 BGP-Server

Name		IP	Nat IP	GRE	Berlin A	Berlin B	Düsseldorf A	Düsseldorf B
Fichtenbackbone-1	IPv4	164.132.13.11	185.66.195.55	remote	100.64.4.40	100.64.4.44	100.64.4.42	100.64.4.46
				lokal	100.64.4.41	100.64.4.45	100.64.4.43	100.64.4.47
	IPv6			remote	2a03:2260:0:21f::1	2a03:2260:0:21f::1	2a03:2260:0:21f::1	2a03:2260:0:21f::1
				lokal	2a03:2260:0:21f::2	2a03:2260:0:21f::2	2a03:2260:0:21f::2	2a03:2260:0:21f::2
Fichtenbackbone-2	IPv4	51.255.115.97	185.66.195.52	remote	100.64.2.125	100.64.2.126	100.64.2.128	100.64.2.131
				lokal	100.64.2.124	100.64.2.127	100.64.2.129	100.64.2.130
	IPv6			remote	2a03:2260:0:14a:1	2a03:2260:0:14a:1	2a03:2260:0:14a:1	2a03:2260:0:14a:1
				lokal	2a03:2260:0:14a:2	2a03:2260:0:14a:2	2a03:2260:0:14a:2	2a03:2260:0:14a:2
Fichtenbackbone-3	IPv4	164.132.83.63	185.66.195.53	remote	100.64.	100.64.	100.64.	100.64.
				lokal	100.64.	100.64.	100.64.	100.64.
	IPv6			remote	2a03:2260:0:	2a03:2260:0:	2a03:2260:0:	2a03:2260:0:
				lokal	2a03:2260:0:	2a03:2260:0:	2a03:2260:0:	2a03:2260:0:
Fichtenbackbone-4	IPv4		185.66.195.54	remote	100.64.	100.64.	100.64.	100.64.
				lokal	100.64.	100.64.	100.64.	100.64.
	IPv6			remote	2a03:2260:0:	2a03:2260:0:	2a03:2260:0:	2a03:2260:0:
				lokal	2a03:2260:0:	2a03:2260:0:	2a03:2260:0:	2a03:2260:0:

1.2.5 Subdomänen

Server	IPv4	IPv4 eth1	IPv6	IPv4 Range	IPv6 Range	Bssid 2.4	Bssid 5
Altena-1	51.255.115.97		2001:41d0:2:b54	172.17.0.1/16	2a03:2260:120:1000	02:ff:13:37:fe:04	02:ff:13:37:fe:05
Iserlohn-1	5.196.239.99	172.31.254.10	2001:41d0:2:b54	172.16.0.1/16	2a03:2260:120::/56	02:ff:13:37:fe:02	02:ff:13:37:fe:04
Meinerzhagen-1	164.132.13.111			172.18.0.1/16	2a03:2260:120:2000	02:ff:13:37:fe:05	02:ff:13:37:fe:06
Hemer-1	164.132.13.114	172.31.254.1		172.19.0.1/16	2a03:2260:120:3000	02:ff:13:37:fe:03	02:ff:13:37:fe:07
Lüdenscheid-1	164.132.153.247	172.31.254.5		172.20.0.1/16	2a03:2260:120:400::/56		
Neuenrade-1	51.254.4.237	172.31.254.15		172.21.0.1/16	2a03:2260:120:500::/56		
MK-1	164.132.239.117	172.31.254.200		172.30.0.1/16	2a03:2260:120:9900::/56	13:37:fe:99	

1.3 Siegerland

Freifunk im Siegerland

AS	65411
IPv6	2a03:2260:100c::/48

1.3.1 Websites

1.3.2 Team

Nick	Name	Kontakt	Was
OETiger	Flo	ffflo@posteo.de	Hilchenbach
TroTLF	Klaus	dev0@mailbox.org	Freudenberg

1.3.3 Bleche

Community/Blech	1	2
Freudenberg		x
Hilchenbach	x	

1.3.4 BGP-Server

Name		IP	Nat IP	GRE	Berlin A	Berlin B	Düsseldorf A	Düsseldorf B
siegerland1	IPv4	176.31.125.140	185.66.194.16	remote	100.64.4.124	100.64.4.128	100.64.4.126	100.64.4.130
				lokal	100.64.4.125	100.64.4.129	100.64.4.127	100.64.4.131
	IPv6			remote	2a03:2260:0:242::1	2a03:2260:0:246::1	2a03:2260:0:245::1	2a03:2260:0:247::1
				lokal	2a03:2260:0:242::2	2a03:2260:0:246::2	2a03:2260:0:245::2	2a03:2260:0:247::2
siegerland2	IPv4	188.165.231.84	185.66.194.17	remote	100.64.4.132	100.64.4.136	100.64.4.134	100.64.4.138
				lokal	100.64.4.133	100.64.4.137	100.64.4.135	100.64.4.139
	IPv6			remote	2a03:2260:0:248::1	2a03:2260:0:244::1	2a03:2260:0:249::1	2a03:2260:0:24b::1
				lokal	2a03:2260:0:248::2	2a03:2260:0:244::2	2a03:2260:0:249::2	2a03:2260:0:24b::2

1.3.5 Konzentratoren

Name	IPv4	IPv6
ff-si-konz-1.ff-si.ovh	51.254.244.185	
ff-si-konz-2.ff-si.ovh	188.165.115.76	

1.3.6 Subdomänen

Server	IPv4	IPv6	IPv4 Range	IPv6 Range	Bssid 2.4	Bssid 5
freudenberg-1.ff-si.ovh	178.33.39.219		172.17.0.1/16	2a03:2260:100c:200::/56		
hilchenbach-1.ff-si.ovh	51.254.244.186		172.16.0.1/16	2a03:2260:100c:100::/56		

1.4 Neanderfunk

1.4.1 Übersicht

Freifunk im Neanderland (Kreis Mettmann).

AS	64863
IPv6	2a03:2260:300e::/48

1.4.2 Websites

Karte	https://karte.neanderfunk.de/
Firmware	https://download.ffnef.de/firmware/
Webseiten	http://www.neanderfunk.de/
	http://www.freifunk-mettmann.de/
	http://freifunk-ratingen.de/
	http://www.freifunk-velbert.de

1.4.3 Team

Nick	Name	Kontakt	Was
Lutz	Lutz Wulfestieg	Lutz@forum	Community, Organisation
Benedikt_Wi	Benedikt	Benedikt_Wi@forum	Firmware, Supernodes
plaste	Stephan	plaste@forum	Dokumentation, Supernodes

1.4.4 Bleche

Community/Blech	1	2	3
Erkrath (erk)	x	x	
Haan (han)	x	x	
Hilden (hld)	x	x	
Heiligenhaus (hlg)	x	x	
Langenfeld (lgf)	x	x	
Mettmann (met)	x	x	
Monheim (mon)	x	x	
Ratingen (rat)	x	x	
Velbert (vel)	x	x	
Wülfrath (wlf)	x	x	

Stand: 29.05.2016; Blech 1/2 sind VM's beim FFRL, keine dezidierten VM's. Blech 3 (FF-ME-Blech1) befindet sich aktuell im Aufbau.

1.4.5 BGP-Server

Name		IP	Nat IP	GRE	Berlin A	Berlin B	Düsseldorf A	Düsseldorf B
neander-1	IPv4	151.80.11.200		remote	100.64.3.86	100.64.3.90	100.64.3.88	100.64.3.92
				lokal	100.64.3.87	100.64.3.91	100.64.3.89	100.64.3.93
	IPv6			remote	2a03:2260:0:1b5::1	2a03:2260:0:1b7::1	2a03:2260:0:1b6::1	2a03:2260:0:1b8::1
				lokal	2a03:2260:0:1b5::2	2a03:2260:0:1b7::2	2a03:2260:0:1b6::2	2a03:2260:0:1b8::2
neander-3	IPv4	151.80.11.203		remote	100.64.3.102	100.64.3.106	100.64.3.104	100.64.3.108
				lokal	100.64.3.103	100.64.3.107	100.64.3.105	100.64.3.109
	IPv6			remote	2a03:2260:0:1bd::1	2a03:2260:0:1bf::1	2a03:2260:0:1be::1	2a03:2260:0:1c0::1
				lokal	2a03:2260:0:1bd::2	2a03:2260:0:1bf::2	2a03:2260:0:1be::2	2a03:2260:0:1c0::2
FF-ME-Blech1	IPv4	37.59.64.72		remote	100.64.3.94	100.64.3.98	100.64.3.96	100.64.3.100
				lokal	100.64.3.95	100.64.3.99	100.64.3.97	100.64.3.101
	IPv6			remote	2a03:2260:0:1b9::1	2a03:2260:0:1bb::1	2a03:2260:0:1ba::1	2a03:2260:0:1bc::1
				lokal	2a03:2260:0:1b9::2	2a03:2260:0:1bb::2	2a03:2260:0:1ba::2	2a03:2260:0:1bc::2

1.4.6 Konzentratoren

Name	IPv4	IPv6
ff-me-blech1.ffnef.de	37.59.64.72	

1.4.7 Subdomänen

Server	IPv4	IPv6	IPv4 Range	IPv6 Range	Bssid 2.4	Bssid 5
TBD						

1.5 Düsseldorf-Flingern

1.5.1 Übersicht

Freifunk-Flingern ist ein Düsseldorfer Projekt welches als Ziele sich den bau von möglichst stark vernetzten Meshes gesetzt hat. Es sollen möglichst große Mesh-Wolken gebaut werden, die vergleichsweise wenige (aber kräftige) VPN-Uplinks haben. Bei der Versorgung von Geflüchtetenunterkünften ist die Nutzung von bestehender Freifunk-Technik (im Rahmen des PPA und des MoU) Zielvorgabe, um nicht nur als "Graswurzel-Internetprovider" aufzutreten, sondern echten Freifunk zu den Refugees zu bringen.

1.5.2 Websites

Karte	http://map.ffdus.de/
Firmware	http://images.ffdus.de/
Projekt	http://www.twin.world/
Blog	http://www.ffdus.de

1.5.3 BGP-Server

AS	65125

Table: broken!

Name		IPvserver	IPffrl	GRE	Berlin A	Berlin B	Düsseldorf A	Düsseldorf B
Flingern-1	IPv4	51.255.150.68	185.66.195.64	remote	100.64.2.200	100.64.2.202	100.64.2.204	100.64.2.206
				lokal	100.64.2.201	100.64.2.203	100.64.2.205	100.64.2.207
	IPv6			remote	2a03:2260:0:16e::1	2a03:2260:0:16f::1	2a03:2260:0:170::1	2a03:2260:0:171::1
				lokal	2a03:2260:0:16e::2	2a03:2260:0:16f::2	2a03:2260:0:170::2	2a03:2260:0:172::2
Flingern-2	IPv4	5.196.239.99	185.66.195.65	remote	100.64.4.40	100.64.4.44	100.64.4.42	100.64.4.46
				lokal	100.64.4.41	100.64.4.45	100.64.4.43	100.64.4.47
	IPv6			remote	2a03:2260:122::1	2a03:2260:0:21e::1	2a03:2260:0:21d::1	2a03:2260:0:21f::1
				lokal	2a03:2260:122::2	2a03:2260:0:21e::2	2a03:2260:0:21d::2	2a03:2260:0:21f::2

1.5.4 Subdomänen

Server	IPv4	IPv6 int	IPv6 ext	nextnode v4	nextnode v6	DHCP v4
w0-9	10.155.0.0/20	fda0:747e:ab29:9375::/64	2a03:2260:122::/48	10.155.0.1	fda0:747e:ab29:9375::1	10.155.11.0-10.155.7.255

1.5.5 Team

Nick	Name	Kontakt	Was
Trickster	Silas	trickster@forum	Geld, Logistik, Unterkünfte, Routerdaten
mst	mathias		Uplinks, Offloader, Neurouter
Frankth	Frank		Dokumentation, Coaching
Adorfer	Andreas	adorfer@forum adorferen@gmail.com	Firmware, Ourdoorinstallationen

1.6 GL.Eulenfunk

1.6.1 Übersicht

GL.Eulenfunk ist eine Untergruppe von Freifunk GL (Freifunk im Rheinisch-Bergischen Kreis). Dort sind die Städte Leichlingen und Bergisch Gladbach mit eigenen Supernodes vertreten, während die anderen Städte in der Domäne GL.Wupper Resourcen beziehen.

1.6.2 Websites

Karte	https://map.ffgl.eu/
Firmware	http://firmware.ffgl.eu/
Webseite	https://freifunk-leichlingen.net/

1.6.3 Team

Nick	Name	Kontakt	Was
Frank	Frank	frank@forum	Eisen, Neurouter
PetaByteBoy	Milan	petabyteboy@forum	Eulenfunk-Karten, VMs

CHAPTER 2

Infrastruktur

2.1 gemietete Hosts

name	owner	hosterloc	typ	os	FQDN	IPv4 (base)	ipv4 (pool)	IPv6	MAC
dags1	Silas	OVH RBX	6M-4C8T-32G-2x2T-500M	PM4	dags1.ffdus.de	5.1[illegible]254.47.[illegible]39	96.175.52/30 51.255.150.68/30	2001:41d0:100[illegible]8:07ef::/64	
paz	Sabine	SYS RBX	4M-4C8T-32G-2x2T-250M	PM4		46.105.121.[illegible]2	51.255.233.208/29	2001:41d0:2:e8d1::/64	
vpn	An-dreas	NC	V-2C-6G-112G-100M	arch		37.120.171.253		2a03:4000:6:5[illegible]:bb:f0:15:12	52:54:[illegible]
ffgek0	An-dreas	NC	V-1C-2G-40G-100M	arch		46.38.238.147		2a03:4000:2:83[illegible]:27:00:19:46	52:54:[illegible]
ff-dus0	An-dreas	NC	V-1C-2G-30G-100M	arch		46.38.234.225		2a03:4000:2:bb[illegible]:cc:64:88:af	96:00:[illegible]
pbbpg	an-dreas	nc	v-2c-6g-230g-100m	arch		5.45.96.247		2a03:4000:5:11[illegible]:a2:c2:e5:f8	ca:03:[illegible]
sil-ver	Frank	SYS		PM4	sil-ver.ffgl.eu	188.165.19[illegible]68	164.132.31.112/30	2001:41d0:2:8b44::/64	

2.2 Virtuelle Hosts

FQDN	host	os	RAM	HDD	mac	ipv4
map.eulenfunk.de	dags1	arch			02:00:00:d6:a0:10	51.255.150.71
map2.eulenfunk.de 0	paz	arch			02:00:00:06:2c:d0	51.255.233.214
flingern-3.ffdus.de	paz	lts14.2			02:00:00:58:04:81	51.255.233.215
iserlohn-2.freifunk-mk.de	dags1	lts14.2			02:00:00:69:ee:4b	5.196.175.52
flingern-1.ffdus.de	dags1	lts14.2			02:00:00:a1:81:5f	51.255.150.68
neander-1.ffnef.de	ffrl	lts16.4	1G	16G	00:50:56:07:e1:83	151.80.11.200
neander-3.ffnef.de	ffrl	ub15.10	1G	16G	00:50:56:0c:29:6f	151.80.11.203
service.ffdus.de	dags1	ipfire			02:00:00:d6:2e:36	51.255.150.70
horst.ffdus	LES	arch	33G	500G	36:e2:87:5e:a4:87	10.155.6.112
konzentrator.silver.ffgl.eu	silver	lts14.2			00:50:56:06:5a:33	164.132.31.113
bgl0.ffgl.eu	silver	lts14.2			00:50:56:07:52:60	164.132.31.114
lln0.ffgl.eu	silver	lts14.2			00:50:56:0a:fa:e8	164.132.31.115

CHAPTER 3

Public Keys (people)

3.1 ssh

Pubkeys, um ssh-Login für die genannten Personen zu einzuräumen:

user	key
pandur	ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAIEAmLQ1QW341TIu6csTplCM1xAKpU8uRLCbcDcQb3P2coBj993PMYhmTwV Pandur
sunta	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC3SPq75cC/tZJ9sWKHXIs1XOuzwc1oIOHzn2TrfpNab5AOZDZ1bXnbn cw@bianca
math-ias	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDBgeaL3Fap41iwHYPG/7khPINzeknkdvtoPWXJJrhTyUGwA/3RbRYleK mathias
lin-gling	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCzSsGrVSdBhrQb94S2fqtUUbwi3aYJnRAGcu2CTiVCNoSlEKYbjXK0A ling@ling
pbb	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDGwGwwqw9qSoEq8L+r7U+FRkGN30iBBA2ohc+fnJK7qSCJ3T9bWJz petabyteboy@pbb-e3
adorfer	ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAIEAufGE6eK/iZTeLKCduy4UEyQpkXX+Z/0SAbYCbkJhJjNpYJaAaMhPI9yPWq adorfer
Benedikt Wi	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC9o5jgZqeWNYM7hpoEnbsCEbKI2NDqIXlUKk6Ty8ftXsvY2MUy9cAk benedikt@kampo
julian	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC+/qANS2vRvc8w7d2v6avmzE0LvzEzQrcjNAKDeo490g3GbnnsuS8Z4F x@x
plaste	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCvMCy/RjnGc/ppzxIqvyv3WnaQ3Tj495MLw8qV4FpTkVndjIuqrY7heW plaste
Wurst-brot	ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEAnrO+trdNZO5/S4tBRXgWJGKljz4DMB9YODxqm7HiHZZC2RuMZaAo5cer wurstbrot
faithin-chaos	ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEAj/IcUJ8h4RrgEwrBc0QIYs53pS5sdQnAC9b+7Q31h0EIL3PwKj3eWs2fQpXD faithinchaos
robot	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDQjdRIoLwaN3DdV7jusUDlLfUK4etdS6zhBdpJb5vFxtT8/sb6+iIvQzrvs ffdusrobot
tuennes	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC3E0csrsSpyqSPpTkrIt1izfS78BLoMNMI6FYa7l+ndi9Xuo5SLltBBhRwl frank@frank-1005HA
domfi	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCC3mQ7wQe+ERDAb6VpnxTrfp+NvG23ItVUy+cjfSCHhJnW7dV8tbt9l domfi
oetiger	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCweucLWW6+o0mZyb4QlvdnGa8AK53G5Q7sCD5XUhrwKBdyf0So3Q flo@flow
Lutz	ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEAlZ+7CFailoU29vuUydvKI3DA6zXB7q7MB6cyv8uv7cERc4JFsHQ/v+Pk7wn lwulfe@freifunk-mettmann.de
trotlf	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDYKm98ZO2xdX2WUtStu6pTS/qCA3idw/Iu4mCcaaz7Ec0SJ4oupIL+we ks@ks-ThinkPad-T430

3.2 firmware

Signkeys für gluon-Firmware/Autoupdater

key	user
‘19a02dd7b50ffc2b59e2cd1f9e76b26b46e33c43ebf641554572e4a677af35cc’, – SenorCafe	
‘2aa020a8860e5c3f638ed77d313148c7e7c5899be4bebf2cb3406875ef03e835’, – Goldwaage	
‘fb919d4adc69bd404f5093ce6b43badf351f9e642ad458406be986baf6096247’, – PetaByteBoy	
‘dd6a9d1aefc175f885705691498e904cbda12cc4602316f04816d78026c7c0f0’, – Benedikt Wi	
‘2a61930930a240c027f6ca4197203d400b6e4a32f9e92041e5f086907796c9bc’, – adorfer	
‘d02f8e60fb7a5069556500694ebe512b6017b01e9950476e4cfcf10d5130c296’, – JJX	
‘7afe187ceb34e83b2cb33c244ab5c8a7e80829c3e83b8d3fc471d2642eb6a602’, – limlug	
‘610e9acf4d550c3a272b88ec5b4cf0a0e382be203f98b860181fb1bcb1641abd’, – mathias	
‘01aff79cb3079b5b343cdc099a342434f284329890230e0f23850a488570b8c2’, – AKA47	
‘579de7b1ded1dc39583515f722d72524f6dce78da635a7ac2d11cfe1dc046e7e’, – tuennes	
‘d0647b68ff46d35394d99630f9337e39786f51f5bdcee5bd26b7b2c729045cb6’, – jenkins	
‘a2e21ef5743945befa8f88f10a1a168050909d82e4d836bcf879eb573b9ce777’, – domfi	
‘96d644ff1ce07d6f67d9329a0eb9a1548d0d01a3519d17ec1fe9d49da3270bfc’, – plaste	

CHAPTER 4

Firmware

4.1 Flingern

TREE	Gluon	status	releasename	date	url
Stable	2015.1.2	nur Wupper Server	20150911-stable	2015-09-11	http://images.ffdus.de/stable/
beta	2015.1.2	nicht nutzen!	20151009-beta	2015-10-09	http://images.ffdus.de/beta/
Experimental	2015.2pre	MoW defekt	2016011502-exp-ssid	2016-01-15	http://images.ffdus.de/experimental/
Broken (nightly)	2016.1pre	untested packates	2016020403-exp	2015-02-04	http://images.ffdus.de/broken/

Stand 2016-02-01:

Vorarbeiten am Gluon Releasekanidaten herumgeschraubt, um beim offizielen Release schnellstmöglich auch eine stable bringen zu können.
 Ein Testkanidat von Mitte Januar hat erhebliche Probleme mit "MoW" wenn zur Bootzeit kein Ethernetlink auf LAN(!) besteht.

Was in der Release dazukommt:

1. einen wöchentlichen Reboot (freitag morgens zwischen 3 und 5)
2. einen verbesserten Wifi-Powerfix (einige Router haben beim Neuflaschen

etwa 3dB zu wenig Sendeleistung eingestellt. Das ist ein Bug aus dem ChaosCalmer. Da werden die RegDomain-Limits leider nicht augeschöpft)

3. eine "Vorrüstung" für "Clientnetz tagsüber 'aus"' oder "Clientnetz über Nacht 'aus"'.

Müssen nur zwei uci-statements abgesetzt werden, dann ist das aktiv. Inhaltlich mag ich es nicht, aber es ist für mich das kleinere Übel im Vergleich zu "gar kein Freifunk an Standort x"

4. eine "Vorrüstung" für "Bandbreiten-Limit Zeitschaltuhr für den VPN-Link", ebenfalls wahlweise tagsüber oder nachts.

(Frage in die Runde: soll ich zu c und d noch was bauen, dass man zwischen Mo-Fr & Sa-So unterschiedlich setzen kann? Irgenwann wäre dann eine Seite im Advanced-Websetup sinnvoll. Falls sich jemand berufen fühlen sollte, den Kampf mit lua-scripts aufzunehmen.)

5. den Wifineighborcheck, um hängende CPE210er (und andere) bei sonst unerkennbarem Verlust des Wifimoduls neu zu starten.

Was ich trotz diverser Versuche nicht hinbekomme und wo Petabyteboy mir aus der Patsche helfen darf ist der "Radiochannel-Keep": Damit die einmal verstellen Frequenzen beim Update nicht wieder auf Default stehen.

(Damit die Kabelmesh-Nodes in den Unterkünften mit ihren verteilten Wlan-Kanälen nicht nach einem Update komplett händisch nachgepflegt werden müssen.)

CHAPTER 5

Server Kochbuch

Anleitung zur Einrichtung eines Freifunk Supernodes auf Basis von Proxmox 3.6 und Ubuntu Server 14.04.4 LTS

Das Setup besteht im wesentlichen aus 3 Zonen:

- Die Freifunk Zone vor Ort (rosa): damit haben wir nicht viel zu tun.
- Die Serverzone (grün): um die geht es in dieser Anleitung.
- Das Backbone (orange), das macht der FFRL.

Die Serverzone teilt sich wiederum in 3 Segmente auf:

- Der Hypervisor Proxmox, dieser stellt alle Funktionen für den Betrieb von virtuellen Maschinen bereit.
- Der Konzentrator, dieser virtuelle Server stellt die Verbindung zu FFRL Backbone her, und übernimmt NAT und BGP.
- Der Supernode stellt die Fastd VPN Verbindungen für die Router bereit, kümmert sich um Batman, DHCP und radvd.

Vom Client ins Internet gehen die Daten folgenden Weg (IPv4):

Und das ist der Rückweg (IPv4):

Weiterlesen:

5.1 Das Blech

Wir treiben ziemlich fiese Dinge mit unseren Servern; CPU und Netzwerktraffic (Volumen und Pakete-Rate) sind die entscheidenden Faktoren.

5.1.1 Voraussetzungen

Folgendes sollte euer Wunschserver leisten, damit er für FF tauglich ist:

- 100 Mbit garantierte Bandbreite
  - Nicht 100 Mbit Anbindung oder Peak Bandbreite! Der Server muss 24/7 100 Mbit abkönnen.

- Vollen Zugriff aufs Blech / den Hypervisor
  - Server neu installieren
  - Hardware reboot
  - IPs hinzufügen
- Leistungsstarke CPU
- Zusätzliche IPv4 Adressen (Failover IPs) “zu Einmal-Kosten”

Folgendes sind absolute Ausschlusskriterien für einen Server:

- Trafficbegrenzung
- Fair use
- Vserver

5.1.2 Hoster / Rechenzentrum

OVH bzw. deren preiswertere Marke “Soyoustart” (sys) sind gut geeignet.

Die auf “OVH” gebrandeten Server leisten kaum mehr als die SYS Maschinen, kosten aber unverhältnismäßig viel mehr.

5.1.3 Bestellvorgang

Wer bei OVH oder SYS die Server bestellt sollte, wenn er Neukunde ist, per Überweisung zahlen. Das erspart zusätzliche Authentifizierungsmethoden und geht daher schneller(!) als Paypal.

5.1.4 Sicherheit

Über das OVH/SYS Kundeninterface hat man die Möglichkeit den Server neu zu starten, neu zu installieren oder in den RescueMode zu booten. Man sollte daher dringend den Zugang zum Kundeninterface mit einer Two-Factor Login Methode zusätzlich absichern. Man kann die selbe OATH App nutzen, die man auch für Github und später das Proxmox Webinterface verwenden kann.

Für iOS wurde z.B. die App “OTP Auth” getestet. Auf Ubuntu-Phone lässt sich die App “Authenticator” einsetzen.

5.2 Proxmox

5.2.1 Einleitung

Proxmox stellt alle Funktionen für den Betrieb von virtuellen Maschinen bereit und bietet per Webinterface eine zentrale Möglichkeit, neue VMs anzulegen und bestehende zu verwalten, inkl. einer KVM-Konsole für VMs und auch den Host selbst. Das funktioniert ohne Spezial-Plugins (d.h. kein Flash, keine JRE etc.)

Die Einrichtung des Proxmox beschränkt sich auf folgende Punkte:

- Installation: Hoster wie OVH/Soyoustart nehmen euch die Arbeit ab
- Einrichtung des SSH Zugriffs per Public-Key
- Absicherung des SSH Servers

- Absicherung des Webinterfaces per Two-Factor-Authentication (Oath)
- Einrichtung des Monitorings per Check_MK
- Bereitstellung der ISO Datei für Ubuntu Server

5.2.2 Installation

Proxmox kommt entweder per Klick als Template vom Provider auf den Server oder muss von Hand installiert werden.

Hier ein Beispiel für die Installation über das Soyoustart Web-Frontend. Zunächst den passenden Server im Dropdown-Menü auswählen und dann auf "Installieren" klicken:

Im nächsten Schritt wählt man VPS Proxmox VE 3.4 (64Bit) als Template aus, der Haken bei "Personalisierte Installation" darf NICHT gesetzt sein. Mit Klick auf "Weiter" startet man jetzt die Installation.

Wenn die Installation seitens Soyoustart abgeschlossen ist, bekommt man eine Benachrichtigung per Mail.

5.2.3 SSH

Im laufenden Betrieb erfolgt die komplette Konfiguration über das Webinterface, trotzdem ist es wichtig, sich für Notfälle einen SSH Zugriff einzurichten und natürlich auch den SSH Server abzusichern.

Per SSH mit dem Server verbinden

```
ssh root@111.222.333.444
```

Kennwort ändern

Wenn Proxmox durch den Hoster aufgesetzt wurde und das Kennwort per Mail kam, sollte es geändert werden mit passwd

```
passwd
```

Normalen Useraccount anlegen

Als zusätzliche Sicherheitsstufe wird der direkte root-Login per ssh komplett untersagt. Der Login erfolgt dann über einen zusätzlich anzulegenden Benutzer. Dieser Benutzer muss über ein sicheres Passwort (TM) abgesichert werden. Nötige administrative Tätigkeiten werden mit sudo ausgeführt.

sudo installieren:

```
apt-get install sudo
```

Neuen User anlegen:

```
useradd meinbenutzername
```

Den neuen User der Gruppe "sudo" hinzufügen:

```
    gpasswd -a meinbenutzername sudo
```

```
cd /home/meinbenutzername/
mkdir .ssh
nano .ssh/authorized_keys
```

Im Editor dann den Public Key ("ssh-rsa AAA.....") einfügen. Wichtig: Alles von diesem Key muss in eine Zeile. Jeder Administrator bekommt nach dem beschriebenen Verfahren seinen eigenen Account.

Nun das Password-Login auf dem Server deaktivieren. Dazu die sshd_config editieren:

```
nano /etc/ssh/sshd_config
```

Die Zeile

```
#PasswordAuthentication yes
```

ändern in

```
PasswordAuthentication no
```

Achtung, auch wenn 'yes' auskommentiert ist, besteht die Möglichkeit sich per Password zu verbinden, erst wenn 'no' gesetzt ist und nicht (mehr) auskommentiert ist, ist der Zugriff nur noch per Key möglich.

Um es den Script-Kiddies und Bots etwas schwerer zu machen, sollte der Port 22 auf einen hohen Port (mindestens über 1024) verändert werden. Dazu die Zeile

```
Port 22
```

ändern z.B. in

```
Port 62954
```

WICHTIG: Diesen Port muss man sich dann merken, da man ihn später beim Aufruf von ssh angeben muss.

Nun den direkten Rootlogin sperren.

```
PermitRootLogin yes
```

ändern in

```
PermitRootLogin no
```

Danach den Editor wieder verlassen und den SSH Server neu starten um die Einstellungen zu übernehmen.

```
/etc/init.d/ssh restart
```

Den nachfolgenden ssh Kommandos muss man die Option "-p 62954" (kleines "p"!) und den scp Kommandos die Option "-P 62954" (großes "P"!).

```
ssh -p 62954 meinbenutzername@111.222.333.444
```

5.2.4 Updates einspielen

Nun Betriebsystemupdates einspielen und ggf. erfolgende Rückfragen mit einem "J" oder "Y" abnicken, das "autoremove wird nicht viel tun, aber der Vollständigkeit halber sollte man es sich gleich angewöhnen.

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get autoremove
```

Eine Fehlermeldung im Bereich "Proxmox-Enterprise" kann man entweder ignorieren. Das gibt es nur wenn man ein Support-Abo abgeschlossen hat. Wenn Ihr die Arbeit des Proxmox-Teams unterstützen möchtet:

https://www.proxmox.com/de/proxmox-ve/preise

Optional:

Da einzelne Repositories wiederholt nicht oder sehr schlecht per IPv6 erreichbar sind und wir unsere Maschinen grundsätzlich zur IPv6-Nutzung befähigen, empfiehlt es sich, IPv6 zumindest für "apt-get" zu unterbinden.

Dazu wird einmalig aufgerufen:

```
sudo ...!! SDFSDF echo 'Acquire::ForceIPv4 "true";' > /etc/apt/apt.conf.d/99force-ipv4
```

5.2.5 Monitoring

Den Check_MK Agent steht in der Weboberfläche des Check_MK als .deb Paket bereit:

In die CheckMK-Instanz per Webbrowser einloggen. Dann suchen:

```
-> WATO Configuration (Menü/Box)
-> Monitoring Agents
-> Packet Agents
-> check-mk-agent_1.2.8p1-1_all.deb _(Beispiel)_
```

Den Download-Link in die Zwischenablage kopieren. Im SSH-Terminal nun eingeben: (die Download-URL ist individuell und der Name des .deb-Paketes ändert sich ggf.)

```
wget --no-check-certificate "https://monitoring.eulenfunk.de/heimathoster/check_mk/agents/check-mk-ag
```

Um das .deb Paket zu installieren wird gdebi empfohlen, ausserdem benötigt der Agent xinetd zum ausliefern der monitoring Daten. Die Installation von gdebi kann durchaus einige Dutzend Pakete holen. Das ist leider normal. Per SSH auf dem Server. (Auch hier: Name des .deb-Files ggf. anpassen)

```
sudo apt-get install gdebi-core xinetd
```

Rückfragen ggf. mit "J" beantworten. Mit dem nun installierten gdebi das check_mk-Paket installieren:

```
sudo gdebi check-mk-agent_1.2.8p1-1_all.deb
```

Nun noch zusätzliche Check_MK Plugins hinzufügen

```
cd /usr/lib/check_mk_agent/plugins
sudo wget --no-check-certificate "https://monitoring.freifunk-mk.de/heimathoster/check_mk/agents/plug
sudo chmod +x smart

                        cd /usr/lib/check_mk_agent/local
sudo wget --no-check-certificate https://raw.githubusercontent.com/eulenfunk/check_mk/master/proxmox
sudo chmod +x proxmox
```

:: sudo nano /etc/xinetd.d/check_mk

Dort die Zeile

```
# only_from = 127.0.0.1 10.0.20.1 10.0.20.2
```

ändern in

```
only_from = 127.0.0.1 94.23.160.148
```

Damit diese Änderungen aktiviert werden, muss der xinetd durchgestartet werden

```
sudo /etc/init.d/xinetd restart
```

Der Rechner hält ab nun Daten zum Abruf bereit.

Eulenfunker müssen dann das Admin-Team kontaktieren, damit der Rechner im CheckMK eintragen wird.

5.2.6 Images hochladen

ISO Files zur installation können zwar über das Webinterface hochgeladen werden, aber je nach Internetanbindung dauert das lange. Per wget wird das Image direkt auf den Server geladen.

```
cd /vz/template/iso
wget http://releases.ubuntu.com/14.04.4/ubuntu-14.04.4-server-amd64.iso
```

5.2.7 OATH Two Factor

Der Zugang zum Proxmox ist absolut sicherheitskritisch, wer Zugriff auf den Hypervisor hat hat Zugriff auf alle Maschinen auf dem Blech. Daher muss zusätzlich der Login des Webinterface per OATH Two Factor Authentifizierung abgesichert werden.

-> https://pve.proxmox.com/wiki/Two-Factor_Authentication

5.2.8 Netzwerk einrichten

Ab jetzt geht die Konfiguration über das Proxmox Webinterface im Browser:

```
https://111.222.333.444:8006
```

Beim ersten Aufruf sollte man das Zertifikat im Browser dauerhaft akzeptieren.

Die Anmeldung erfolgt mit Benutzername, Kennwort und OTP Pin. Als Realm muss Linux PAM standard authentication (+ oath) ausgewählt werden.

Nachdem links in der Seitenleiste das Blech ausgewählt wurde rechts im Reiter Network zusätzlich zur vorhandenen vmbr0 über die das Internet rein kommt noch mindestens eine vmbr1 anlegen, über die die Supernodes mit dem Konzentrator kommunizieren.

Bei OVH/Soyoustart kann es sein, dass die vmbr schon vorhanden ist, dann müsst ihr nichts tun.

Beim Anlegen muss als Name vmbr1 eingetragen werden und der Haken bei Autostart gesetzt werden.

Die vmbr steht erst nach dem Neustart des Blechs zu Verfügung, daher in der Ecke oben rechts “Restart” auswählen.

5.2.9 Backup anlegen

Proxmox ermöglicht es ganz einfach und auf Wunsch automatisiert Backups von den Virtuellen Maschinen anzulegen. Im Idealfall sollten die Backups auf einen externen Server/Storage erfolgen. Aus Gründen der Einfachheit beginnen wir mit einem Backup auf den lokalen Storage. Von dort kann man die Dateien für den Fall eines Totalausfalls des Blechs bei Bedarf per scp oder rsync auf einen anderen Server oder den heimischen Computer sichern.

Das Backup auf dem lokalen Storage erzeugt massiv IO, denn neben den normalen Zugriffen, die die Maschinen im Betrieb erzeugen kommen noch Lesezugriffe auf die zu sichernde VM und Schreibzugriffe auf die Backupdatei dazu.

Sobald der IO die Kapazität des Storages übersteigt, gerade bei den einfachen Raids aus klassischen HDDs in den OVH/SYS Servern ist dies schnell der Fall, wird die Performance des gesamten Blechs und aller VMs darunter leiden.

Das Backup sollte daher zur Zeit der geringsten Auslastung erfolgen, z.B. jeden Montag um 1 Uhr in der Nacht.

Zuerst muss ein Backupstorage definiert werden, dazu muss links das Datacenter ausgewählt werden, rechts der Tab Storage und dort der lokale Storage konfiguriert werden.

Dort muss dann VZDump backup file zusätzlich ausgewählt werden (STRG+Klick)

Als nächstes im Reiter Backup einen Backupjob hinzufügen. Bei Node wird "– All –" und bei Mode Snapshot ausgewählt. Storage setzt man auf local. Als Compression wählt man "LZO (fast)" um die Prozessorauslastung gering zu halten.

5.3 BGP Konzentrator einrichten

Der BGP Konzentrator ist der Backboneseitige unserer zwei Freifunk Server, er übernimmt Routing, NAT, Connection tracking, GRE Tunnel und BGP Sessions.

Für die virtuelle Maschine benötigen wir eine öffentliche IPv4 Adresse. Diese könnt ihr beim Rechenzentrum kaufen, nennt sich z.B. Failover IP. Für diese IP Adresse muss im Kundeninterface eine MAC Adresse erstellt werden, die dann im Proxmox auf der Netzwerkkarte der virtuellen Maschine konfiguriert wird.

Im Kundeninterface wird "IP" ausgewählt

In dem erscheinenden Formular klickt man das Zahnrad an der betreffenden IP-Adresse an und wählt "Eine virtuelle MAC-Adresse hinzufügen"

Im folgenden muss "Eine neue virtuelle MAC-Adresse erstellen" angeklickt werden und der Name der VM eingetragen werden.

Auf dem Webinterface des Proxmox Servers ist auf der linken Seite das Blech auszuwählen und dann oben rechts 'Create VM' anklicken

---

Im Reiter 'General' eine Freie ID und einen Namen festlegen.

Im Reiter 'OS' 'Linux 4.x/3.x/2.6 Kernel auswählen.

Im Reiter 'CD/DVD' das ISO Image auswählen.

Im Reiter 'Hard Disk' als 'Bus' 'VirtIO' einstellen, die Festplattengröße auf 6GB begrenzen und als Format 'qcow2' wählen. Größere Festplatten machen Backups, Rollbacks und co nur aufwändiger.

Im Reiter 'CPU' ein Prozessorkern zuweisen. Als CPU kann man "host" wählen, das tut der Performance gut und HA nutzen wir ohnehin nicht.

Im Reiter 'Memory' unter 'Automatically allocate memory within this range' 256 - 1024MB festlegen. Weniger als 256 hindert einige Maschinen beim booten, mehr als 1024 werden nicht benötigt.

Im Reiter 'Network' als Netzwerkkarte 'VirtIO' auswählen und die MAC Adresse der für diese VM zu verwendenden öffentlichen IPv4 Adresse eintragen. Bridged Mode übernehmen wir so und vmbr0 auch diese.

Bestätigen und Anlegen auswählen.

Fehlermeldungen während der Startphase werden unten im Log-Fenster angezeigt, erscheinen immer “oben”, jedoch mit einigen Sekunden Verzögerung. Details lassen sich ausklappen.

Hinweis: Wenn das System später läuft, nicht vergessen, die Option “Start at boot” auf “Yes” zu stellen.

5.3.1 Ubuntu Server Installieren

Die VM links auswählen und oben rechts starten und die Konsole öffnen

---

Deutsch als Sprache auswählen und nun Ubuntu Server Installieren

---

Als Installationssprache jetzt nochmal Deutsch auswählen,

die Auswahl trotz unvollständiger Unterstützung bestätigen,

den Standort auswählen (Deutschland),

das Tastaturmodell nicht automatisch erkennen lassen

Herkunftsland der Tastatur “Deutsch”

Tastaturbelegung “Deutsch”

Sobald der Server versucht das Netzwerk automatisch zu konfigurieren, dies abbrechen und die manuelle Netzwerkkonfiguration auswählen.

Die Failover-IP, für die wir vorhin die MAC-Adresse erstellt haben ist beispielsweise die 555.666.777.888

Die Subnetzmaske von 255.255.255.0 bleibt in der Regel so

Die Gateway Adresse sollte man beim Rechenzentrum erfragen.

Bei OVH/Soyoustart ist das IPv4 Gateway immer auf der 254, also 555.666.777.254

Als DNS geht z.B. der 8.8.8.8 von Google (Böse!).

Der Rechnername ist frei wählbar

Der Domainname ist hier einzutragen

Und der Benutzer angelegt werden. Zunächst der volle Benutzername

und dann das gewünschte Login

Das Kennwort sollte sicher sein und nicht bereits für einen anderen Zweck in Verwendung.

Da auf dem Server keine persönlichen Dateien gespeichert werden sollen ist es nicht notwendig den persönlichen Ordner zu verschlüsseln.

Zeitzone Prüfen und bestätigen.

Festplatte manuell formatieren

Freien Speicherplatz auswählen und enter

Partitionstabelle erstellen

Freien Speicherplatz auswählen und enter

Partitionsgröße 5 GB Primär am Anfang

Bootflag auf ‘ein’ setzen und ‘Anlegen beenden’

Freien Speicherplatz auswählen und enter

Eine neue Partition erstellen

Größe bestätigen

Primär

Benutzen als ‘Auslagerungsspeicher (SWAP)’

‘Anlegen beenden’

‘Partitionierung beenden’

Ja schreiben, noch sind ja keine Daten vorhanden, die überschrieben werden könnten.

Warten...

Proxy leer lassen

Warten...

Automatische Sicherheitsaktualisierungen auswählen

OpenSSH server auswählen (Leertaste benutzen) und weiter

Warten...

Die Installation des GRUB Bootloader bestätigen

Weiter

5.3.2 SSH

Per SSH mit dem Server verbinden

```
ssh meinbenutzername@111.222.333.444
```

Den Public-Key für den User hinterlegen:

```
cd /home/meinbenutzername/
mkdir .ssh
nano .ssh/authorized_keys
```

Im Editor dann den Public Key ("ssh-rsa AAA.....") einfügen. Wichtig: Alles von diesem Key muss in eine Zeile. Weitere Adminuser können später angelegt werden.

Nun das Password-Login auf dem Server deaktivieren. Dazu die sshd_config editieren:

```
sudo nano /etc/ssh/sshd_config
```

Die Zeile

```
#PasswordAuthentication yes
```

ändern in

```
PasswordAuthentication no
```

Achtung, auch wenn 'yes' auskommentiert ist, besteht die Möglichkeit sich per Password zu verbinden, erst wenn 'no' gesetzt ist und nicht (mehr) auskommentiert ist, ist der Zugriff nur noch per Key möglich.

Um es den Script-Kiddies und Bots etwas schwerer zu machen, sollte der Port 22 auf einen hohen Port (mindestens über 1024) verändert werden. Dazu die Zeile

```
Port 22
```

ändern in

```
Port 62954
```

WICHTIG: Diesen Port muss man sich dann merken, da man ihn später beim Aufruf von ssh angeben muss.

Nun den direkten Rootlogin sperren.

```
PermitRootLogin yes
```

ändern in

```
PermitRootLogin no
UsePAM no
```

Danach den Editor wieder verlassen und den SSH Server neu starten um die Einstellungen zu übernehmen.

```
sudo service ssh restart
```

Den nachfolgenden ssh Kommandos muss man die Option "-p 62954" (kleines "p"!) und den scp Kommandos die Option "-P 62954" (großes "P"!).

```
ssh -p 62954 meinbenutzername@111.222.333.444
```

5.3.3 Systemaktualisierung

Als Nächstes steht die Systemaktualisierung an; auch hier beim erstmaligen Aufruf die Nutzung von IPv4 erzwingen für's APT-Get

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get autoremove
```

5.3.4 Pakete installieren

```
sudo apt-get install bird bird6 xinetd vnstat vnstati gdebi-core lighttpd git conntrack
```

- bird übernimmt das BGP routing
- bird6 tut das selbe für IPv6
- vnstat monitort den Netzwerktraffic
- vnstati erzeugt daraus Grafiken
- lighttpd stellt diese zum Abruf bereit
- gdebi-core ermöglicht uns die Installation des Check_mk Agents
- git wird für die Konfigurationsscripte benötigt
- xinetd ist der bei Debian übliche Super-Daemon, über ihn wird der Check_mk Agent angesprochen
- conntrack überwacht den Auslastungszustand der NAT-Engine

5.3.5 Hinzufügen einer Schnittstelle eth1

Für die Verbindung zwischen den Supernodes und dem Konzentrator legen wir eine zweite Netzwerkschnittstelle an. Dazu muss im Proxmox für die VM eine eth1 hinzugefügt werden, die auf der vmbr1 hängt und virtio verwendet.

Danach die VM einmal durchbooten.

5.3.6 Eulenfunk BGP-Konzentrator-Konfigurator

Ist leider noch Baustelle hier... Bis auf weiteres geht es mit unten bei **ferm_einrichten_** weiter.

Die genauen Hintergründe sollten verstanden werden und sind weiter unten beschrieben!

Um die Konfiguration zu vereinfachen, wurde ein Script geschrieben, welches die nötigen Parameter abfragt und daraus die Konfigurationsdateien, bzw. Auszüge daraus erzeugt. Diese müssen dann nur noch an die richtige Stelle kopiert werden.

```
sudo mkdir -p /opt/eulenfunk/konzentrator
cd /opt/eulenfunk/konzentrator
sudo git clone https://github.com/eulenfunk/ff-bgp-konzentrator-konfigurator.git
cd ff-bgp-konzentrator-konfigurator
sudo ./bgp-konzentrator-setup.sh
```

Das Script fragt dann die nötigen Werte ab.

Beschreibung der abgefragten Werte

Allgemeine Parameter

AS Nummer vom FF-RL Hier wird die Nummer des autonomen Systems vom Freifunk Rheinland eingetragen. Aktuell ist das 201701.

Eigene AS Nummer Ihr benötigt ein eigenes autonomes System. Die Nummer davon gebt ihr hier an. TODO: Link auf Beschreibung zur Beschaffung eines eigenen AS...

Zugewiesene FFRL-IPV4-Exit-Adresse Vom Freifunk Rheinland bekommt ihr eine Exit-Adresse. Darauf wird der gesamte IPv4 Verkehr aller an diesem Konzentrator angeschlossenen Supernodes bzw. der darüber verbundenen Clients ge-NAT-ed. Diese Adresse sieht in etwa so aus: 185.66.19X.YY

Zugewiesenes FFRL-IPV6-Netz Der IPv6 Prefix, der euch vom Freifunk Rheinland zugewiesen wurde. (2a03:2260:XXX::/48)

Eigene öffentliche IPV4 Adresse Bei der Einrichtung der VM für diesen Konzentrator habt ihr eine IPv4-Adresse konfiguriert (Failover-IP der VM), über die ihr euch auch auf dem Konzentrator eingeloggt habt. Also die IPv4-Adresse von *eth1*.

Eigener SSH-Port Ihr habt bei der Konfiguration vom *sshd* den Port angepasst (62954), also gebt ihr diesen hier ein. Damit wird sichergestellt, dass die Firewall (ferm ...) Verbindungen zu dem alternativen Port überhaupt zulässt. Wenn ihr euch hier vertut, kommt ihr nach dem Neustart nicht mehr per SSH auf euren Server!

Konfiguration für GRE-Tunnel nach XXX_Y

Ihr solltet vom Freifunk Rheinland Adressen für 4 Tunnel zum Backbone bekommen haben, jeweils zwei in Berlin und zwei in Düsseldorf. In diesem Abschnitt werden diese konfiguriert. Die folgenden Werte müsst ihr jeweils einmal pro Tunnel passend – also 4 Mal – eingeben:

IPV4 Adresse für Tunnelendpunkt auf Backbone-Server Die Tunnel-interne IPv4 Adresse auf dem **Backbone-Server** (100.64.X.YYY gerade).

IPV4 Adresse für Tunnelendpunk auf Konzentrator Die Tunnel-interne IPv4 Adresse für den Tunnelendpunkt auf **eurem Konzentrator** (10.64.X.ZZZ nächste ungerade).

IPV6 Adresse auf Backbone-Server Zusätzlich zu den IPv4-Adressen habt ihr eine IPv6 Adresse für den Tunnel bekommen. Die Adresse mit der ()...)::**1**/64 hinten ist die Adresse auf dem Backbone-Server (in etwa diese 2a03:2260:Y:XXX::1 ohne die /64!). Diese gebt ihr hier an.

IPV6 Adresse auf Konzentrator Die auf die im vorherigen Schritt folgende Addresse, also mit ()...)::**2**/64 hinten, ist die Adresse auf eurem Konzentrator (in etwa diese 2a03:2260:Y:XXX::2 ohne die /64!). Diese gebt ihr hier an.

Ausgaben

Das Script erzeugt folgende Dateien:

- bird.conf.bgp
- bird6.conf.bgp
- interfaces.bgp
- ferm.conf.bgp
- 20-ff-config.conf.bgp

Die erzeugten Dateien sollten nun **überprüft** werden (Beschreibung hierzu siehe unten) und dann an die passenden Stellen kopiert werden:

```
sudo cp bird.conf.bgp /etc/bird/bird.conf
sudo cp bird6.conf.bgp /etc/bird/bird6.conf
sudo mkdir /etc/ferm
sudo cp ferm.conf.bgp /etc/ferm/ferm.conf
sudo cp 20-ff-config.conf.bgp /etc/sysctl.d/20-ff-config.conf
sudo cat interfaces.bgp >> /etc/network/interfaces
```

Da nun ein eventueller alternativer SSH-Port in die ferm.conf eingetragen wurde, kann das Firewalling aktiviert werden.

Als erstes ferm installieren.

```
sudo apt-get install ferm
```

Bei der Frage, ob ferm beim Systemstart gestartet werden soll, mit ja antworten.

Danach kann das System rebootet werden. Die Konfigurationen für die Supernodes werden später wie unten beschrieben angelegt.

Routing

Zum Routing werden Regeln benötigt, die die Pakete aus dem Freifunk Netz und die Pakete vom FFRL Backbone in eine gesonderte Tabelle (Tabelle 42) leiten. In dieser Tabelle wird vom bird per BGP eine Defaultroute ins Backbone gesetzt und manuell Routen zum eigenen Freifunk Netz (zu den Supernodes).

Um eine Menge Handarbeit zu sparen wird das Anlegen der Rules für die einzelnen Communities/Supernodes per Script erledigt.

Das Script gibt es hier: https://github.com/eulenfunk/scripts/tree/master/konzentrator

```
cd /opt/eulenfunk/konzentrator
sudo git clone https://github.com/eulenfunk/konzentrator.git
cd konzentrator
sudo chmod +x *.sh
sudo mkdir config
```

Damit das Script auch beim boot seine Arbeit verrichten kann muss es in die rc.local eingetragen werden.

```
sudo nano /etc/rc.local
```

```
#!/bin/sh -e
# rc.local
/opt/eulenfunk/konzentrator/konzentrator/bgp-konzentrator-rc.sh
exit 0
```

Im Ordner **config** wird je Supernode ein config file angelegt. Die Beschreibung zum Hinzufügen von Supernodes erfolgt im Dokument "Supernode einrichten".

5.3.7 Monitoring

Das Monitoring beinhaltet folgende Komponenten:

- Check_MK ermöglicht das zentrale Monitoring aller Systemdaten aller eingebundenen Server
- vnstat erstellt Traffic Statistiken, die sich auf der shell anzeigen lassen
- vnstati generiert daraus Grafiken
- lighttpd stellt diese zum Abruf aus dem Internet bereit

Check_MK Agent installieren

Den Check_MK Agent steht in der Weboberfläche des Check_MK als .deb Paket bereit:

In die CheckMK-Instanz per Webbrowser einloggen. Dann suchen:

```
-> WATO Configuration (Menü/Box)
-> Monitoring Agents
-> Packet Agents
-> check-mk-agent_1.2.8p1-1_all.deb _(Beispiel)_
```

Den Download-Link in die Zwischenablage kopieren. Im SSH-Terminal nun eingeben: (die Download-URL ist individuell und der Name des .deb-Paketes ändert sich ggf.)

```
wget --no-check-certificate \
https://monitoring.freifunk-mk.de/heimathoster/check_mk/agents/check-mk-agent_1.2.8p1-1_all.deb
```

Um das .deb Paket zu installieren wird gdebi empfohlen, ausserdem benötigt der Agent xinetd zum Ausliefern der Monitoring Daten.

Per SSH auf dem Server. (Auch hier: Name des .deb-Files ggf. anpassen)

```
sudo gdebi check-mk-agent_1.2.8p1-1_all.deb
```

Anschließend noch das Konzentrator-Plugin hinzufügen:

```
cd /usr/lib/check_mk_agent/local
sudo wget -O konzentrator https://raw.githubusercontent.com/eulenfunk/check_mk/master/konzentrator
sudo chmod 755 konzentrator
sudo chmod +x konzentrator
```

Der Rechner hält ab nun Daten zum Abruf bereit.

JJX Bescheid sagen, der kümmert sich dann darum.

vnstat einrichten

Alle 5 Minuten werden die Grafiken der Durchsatzdaten aktualisiert:

```
sudo mkdir -p /var/www/vnstats/eth0
sudo mkdir -p /var/www/vnstats/eth1
sudo nano /etc/cron.d/vnstat
```

```
*/5 * * * * root vnstati -i eth0 -o /var/www/vnstats/eth0/hours.png -h
*/5 * * * * root vnstati -i eth0 -o /var/www/vnstats/eth0/days.png -d
*/5 * * * * root vnstati -i eth0 -o /var/www/vnstats/eth0/months.png -m
*/5 * * * * root vnstati -i eth0 -o /var/www/vnstats/eth0/summary.png -s
*/5 * * * * root vnstati -i eth1 -o /var/www/vnstats/eth1/hours.png -h
*/5 * * * * root vnstati -i eth1 -o /var/www/vnstats/eth1/days.png -d
*/5 * * * * root vnstati -i eth1 -o /var/www/vnstats/eth1/months.png -m
*/5 * * * * root vnstati -i eth1 -o /var/www/vnstats/eth1/summary.png -s
```

5.4 Supernode einrichten

Der Supernode ist der Freifunkseitige unserer zwei Server. Er übernimmt die Adressvergabe per DHCP / Radvd, den Aufbau der Fastd Tunnel zu den Routern und Batman.

Der Supernode wird im Proxmox Webinterface angelegt indem man auf der linken Seite den Server auswählt und dann oben rechts auf 'Create VM' klickt.

Im Reiter 'General' eine Freie ID und einen Namen (meinestadt-1) festlegen.

Im Reiter 'OS' 'Linux 4.x/3.x/2.6 Kernel auswählen.

Im Reiter 'CD/DVD' das ISO Image auswählen.

Im Reiter 'Hard Disk' als 'Bus' 'VirtIO' einstellen, die Festplattengröße auf 6GB begrenzen und als Format 'qcow2' wählen.

Im Reiter 'CPU' einen Prozessorkern zuweisen. Type "host" (mehr Performance, weniger Portabilität).

Im Reiter 'Memory' unter 'Automatically allocate memory within this range' 256-2048MB festlegen.

Im Reiter 'Network' als Netzwerkkarte 'VirtIO' auswählen und die MAC Adresse der für diese VM zu verwendenden öffentlichen IPv4 Adresse eintragen. Bridged Mode übernehmen wir so und vmbr0 auch diese.

Bestätigen und Anlegen, auswählen und anschließend starten.

Fehlermeldungen während der Startphase werden unten im Log-Fenster angezeigt, erscheinen immer "oben", jedoch mit einigen Sekunden Verzögerung. Details lassen sich ausklappen.

Hinweis: Wenn das System später läuft, nicht vergessen, den Starttyp "at boot time" zu stellen und das CD-ROM-Laufwerk entfernen.

5.4.1 Ubuntu Server Installieren

Die VM links auswählen und oben rechts starten und die Konsole öffnen

Deutsch als Sprache auswählen und nun Ubuntu Server installieren

Als Installationssprache jetzt nochmal Deutsch auswählen, die auswahl trotz unvollständiger Unterstützung bestätigen und als nächstes das Tastaturlayout auswählen.

Sobald der Server versucht das Netzwerk automatisch zu konfigurieren, dies abbrechen und die manuelle Netzwerkkonfiguration auswählen.

Die IP zur mac ist beispielsweise die 555.666.777.888

Die Subnetzmaske von 255.255.255.0 bleibt in der Regel so

Die Gateway Adresse sollte man beim Rechenzentrum bekannt sein.

Bei einem großen Französichen RZ ist das IPv4 Gateway immer auf der 254, also 555.666.777.254

Als DNS geht z.B. der 8.8.8.8 von google.

Der Rechnername ist frei wählbar z.b. meinestadt-1

Der Domainname ist hier einzutragen

Und der Benutzername.

Das Kennwort sollte sicher sein und nicht bereits für einen anderen Zweck in Verwendung.

Da auf dem Server keine Persönlichen Dateien gespeichert werden sollen ist es nicht notwendig den persönlichen Ordner zu verschlüsseln.

Zeitzone prüfen und bestätigen.

Festplatte manuell formatieren

Freien Speicherplatz auswählen und Enter

Partitionstabelle erstellen

Freien Speicherplatz auswählen und Enter

Partitionsgröße 5 GB primär am Anfang

Bootflag auf ‘ein’ setzen und ‘Anlegen beenden’

Freien Speicherplatz auswählen und Enter

Einen neue Partition erstellen

Größe bestätigen

Primär

Benutzen als ‘Auslagerungsspeicher (SWAP)’

‘Anlegen beenden’

‘Partitionierung beenden’

Ja schreiben, noch sind ja keine Daten vorhanden, die überschrieben werden könnten.

Warten...

Proxy leer lassen

Warten...

Automatische Sicherheitsaktualisierungen auswählen

OpenSSH Server auswählen (Leertaste benutzen) und weiter

Warten...

Die Installation des GRUB Bootloader bestätigen

Weiter

5.4.2 SSH

Per SSH mit dem Server verbinden

```
ssh meinbenutzername@111.222.333.444
```

Den Public-Key für den User hinterlegen:

```
cd /home/meinbenutzername/
mkdir .ssh
nano .ssh/authorized_keys
```

Im Editor dann den Public Key ("ssh-rsa AAA.....") einfügen. Wichtig: Alles von diesem Key muss in eine Zeile. Weitere Adminuser können später angelegt werden.

Nun das Password-Login auf dem Server deaktivieren. Dazu die sshd_config editieren:

```
sudo nano /etc/ssh/sshd_config
```

Die Zeile

```
#PasswordAuthentication yes
```

ändern in

```
PasswordAuthentication no
```

Achtung, auch wenn 'yes' auskommentiert ist, besteht die Möglichkeit sich per Password zu verbinden, erst wenn 'no' gesetzt ist und nicht (mehr) auskommentiert ist, ist der Zugriff nur noch per Key möglich.

Um es den Script-Kiddies und Bots etwas schwerer zu machen, sollte der Port 22 auf einen hohen Port (mindestens über 1024) verändert werden. Dazu die Zeile

```
Port 22
```

ändern in

```
Port 62954
```

WICHTIG: Diesen Port muss man sich dann merken, da man ihn später beim Aufruf von ssh angeben muss.

Nun den direkten Rootlogin sperren.

```
PermitRootLogin yes
```

ändern in

```
PermitRootLogin no
UsePAM no
```

Danach den Editor wieder verlassen und den SSH Server neu starten um die Einstellungen zu übernehmen.

```
sudo service ssh restart
```

Den nachfolgenden ssh Kommandos muss man die Option "-p 62954" (kleines "p"!) und den scp Kommandos die Option "-P 62954" (großes "P"!).

```
ssh -p 62954 meinbenutzername@111.222.333.444
```

5.4.3 Systemaktualisierung

Als Nächstes steht die Systemaktualisierung an; auch hier beim erstmaligen Aufruf die Nutzung von IPv4 erzwingen für's APT-Get

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get autoremove
```

5.4.4 Pakete installieren

Ergänzen der /etc/apt/sources.list um das fastd repository

```
sudo nano /etc/apt/sources.list
```

Folgende Zeile hinzufügen

```
deb http://repo.universe-factory.net/debian/ sid main
```

Editor schließen

```
sudo apt-get update
sudo apt-get install xinetd git vnstat vnstati gdebi-core lighttpd fastd build-essential \
bridge-utils isc-dhcp-server radvd libnl-3-dev pkg-config
```

Rückfrage mit "J" bestätigen

Um welche Paket handelt es sich?

- vnstat monitort den Netzwerktraffic
- vnstati erzeugt daraus Grafiken
- lighttpd stellt diese zum Abruf bereit
- gdebi-core ermöglicht uns die Installation des Check_mk Agents
- xinetd ist der bei Debian übliche Super-Daemon, über ihn wird der Check_mk Agent angesprochen
- Fastd baut Tunnelverbindungen zu den Routern auf
- build-essential wird zum kompilieren von Batman benötigt
- bridge-utils (brctl) steuert Netzwerkbrücken
- isc-dhcp-server (dhcpd3) verteilt IPv4 Adressen
- radvd verteilt die IPv6 Range
- git wird für die Konfigurationsscripte benötigt
- libnl-3-dev wird für batman benötigt
- pkg-config wird für batctl benötigt

5.4.5 Batman kompilieren

Batman kann man bei http://www.open-mesh.org/projects/open-mesh/wiki/Download herunterladen

```
cd ~
wget http://downloads.open-mesh.org/batman/stable/sources/batman-adv/batman-adv-2016.0.tar.gz
tar -xf batman-adv-2016.0.tar.gz
cd batman-adv-2016.0
make
sudo make install
```

5.4.6 Batctl kompilieren

```
cd ~
sudo wget https://downloads.open-mesh.org/batman/stable/sources/batctl/batctl-2016.0.tar.gz
tar -xf batctl-2016.0.tar.gz
cd batctl-2016.0
make
sudo make install
```

5.4.7 Batman Kernelmodul eintragen

Damit das Batman Kernelmodul beim boot geladen wird müssen wir es noch in die /etc/modules eintragen.

Mehr infos gibt es im ubuntuusers wiki https://wiki.ubuntuusers.de/Kernelmodule#start

```
sudo nano /etc/modules
```

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
batman-adv
```

5.4.8 Fastd einrichten

- Verzeichnis für die Fastd Instanz anlegen
- Dummyverzeichnis für Clients anlegen
- fastd.conf erstellen

```
sudo mkdir -p /etc/fastd/client/dummy
cd /etc/fastd/client
sudo nano fastd.conf
```

```
bind any:10000 default ipv4;
include "secret.conf";
include peers from "dummy";
interface "tap0";
log level info;
mode tap;
method "salsa2012+umac";
peer limit 200;
mtu 1406;
secure handshakes yes;
log to syslog level verbose;
status socket "/run/fastd.client.sock";

on up "
                ip link set address 04:EE:EF:CA:FE:3A dev tap0
                ip link set up tap0
                /usr/local/sbin/batctl -m bat0 if add $INTERFACE
                ip link set address 02:EE:EF:CA:FE:FF:3A dev bat0
                ip link set up dev bat0
                brctl addif br0 bat0
                /usr/local/sbin/batctl -m bat0 it 5000
                /usr/local/sbin/batctl -m bat0 bl 0
```

```
                /usr/local/sbin/batctl -m bat0 gw server 48mbit/48mbit
                /usr/local/sbin/batctl -m bat0 vm server
";

on verify "/etc/fastd/client/blacklist.sh $PEER_KEY";
```

Nun das Blacklist-Script anlegen.

```
sudo nano /etc/fastd/client/blacklist.sh
```

Mit Inhalt:

```
#!/bin/bash
PEER_KEY=$1
echo peer "$PEER_KEY" joining
if /bin/grep -Fq $PEER_KEY /etc/fastd/client/fastd-blacklist.json; then
exit 1
else
exit 0
fi
```

dann die Datei ausführbar machen

```
sudo chmod +x  /etc/fastd/client/blacklist.sh
```

Und schließlich eine Dummy-Datei anlegen

```
sudo nano /etc/fastd/client/fastd-blacklist.json
```

dort hinein

```
{
"peers":
[
{
"pubkey":"0004df72c02827333bced7680acaf38f36b09597c55241571e90637465831000",
}
]
}
```

Den Editor wieder verlassen und nun einen fastd Key erzeugen, der in passender Syntax in “secret.conf” abgelegt wird.

```
sudo fastd --generate-key > secret.conf
```

In der Datei secret.conf müssen dann manuell Änderungen vorgenommen werden: Die Zeile mit ‘Public’ muss mit ‘#’ auskommentiert werden, die Zeile ‘Secret’ muss angepasst werden.

```
sudo nano secret.conf
```

```
secret "xxx...";
#Public: ...
```

5.4.9 Hinzufügen einer Schnittstelle eth1

Nun muss im Proxmox für die VM eine eth1 hinzugefügt werden, die auf der vmbr1 hängt und Virtio verwendet.

Danach die VM einmal durchbooten.

5.4.10 Verbindung zwischen Supernode und Konzentrator konfigurieren

Auf dem Supernode

Zunächst müssen die nötigen Scripte auf den Supernode heruntergeladen und ausführbar gemacht werden:

```
sudo mkdir -p /opt/eulenfunk
cd /opt/eulenfunk
sudo git clone https://github.com/eulenfunk/supernode.git
cd supernode
sudo chmod +x *.sh
sudo chmod +x *.py
```

Nun muss man dem jeweiligen Supernode aus dem vom FFRL zugeteilten IPv6-Adressbereich noch ein /56 herausschneiden, ein passendes IPv4 Netz für seine Endgeräte festlegen und die Werte in die Konfigurationsdatei supernode.config schreiben:

```
sudo nano /opt/eulenfunk/supernode/supernode.config
```

Hier ein Beispiel:

```
SUPERNODE_IPV6_PREFIX=2a03:2260:X:Y::/56
SUPERNODE_IPV4_CLIENT_NET=172.19.0.0/16
SUPERNODE_IPV4_TRANS_ADDR=172.31.254.1
```

Die angepasste Konfiguration wird dann durch das Setup verwendet:

```
cd /opt/eulenfunk/supernode
sudo ./supernode-setup.sh
```

```
Ausgaben in:
        interfaces.eulenfunk
        dhcpd.conf.eulenfunk
        radvd.conf.eulenfunk
        20-ff-config.conf.eulenfunk
```

Die so erzeugten Konfigurationsdateien müssen **nach Prüfung** an die passenden Stellen kopiert werden

```
sudo cp dhcpd.conf.eulenfunk /etc/dhcp/dhcpd.conf
sudo cp radvd.conf.eulenfunk /etc/radvd.conf
sudo cp 20-ff-config.conf.eulenfunk /etc/sysctl.d/20-ff-config.conf
```

und die Netzwerkkonfiguration an die vorhandene angehängt werden:

```
sudo cat interfaces.eulenfunk >> /etc/network/interfaces
```

Als letzter Schritt auf dem Supernode muss die /etc/rc.local folgendermassen angepasst werden:

```
sudo nano /etc/rc.local
```

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
```

```
#
# By default this script does nothing.

/opt/eulenfunk/supernode/supernode-rc.sh

exit 0
```

Das sorgt dafür, dass beim Systemstart durch das Script supernode-rc.sh die nötigen Routen und Routing-Policies konfiguriert werden.

Check_MK Agent installieren

Den Check_MK Agent steht in der Weboberfläche des Check_MK als .deb Paket bereit:

In die CheckMK-Instanz per Webbrowser einloggen. Dann suchen:

```
-> WATO Configuration (Menü/Box)
-> Monitoring Agents
-> Packet Agents
-> check-mk-agent_1.2.8p1-1_all.deb _(Beispiel)_
```

Den Download-Link in die Zwischenablage kopieren. Im SSH-Terminal nun eingeben: (die Download-URL ist individuell und der Name des .deb-Paketes ändert sich ggf.)

```
wget --no-check-certificate \
https://monitoring.freifunk-mk.de/heimathoster/check_mk/agents/check-mk-agent_1.2.8p1-1_all.deb
```

Um das .deb Paket zu installieren wird gdebi empfohlen, ausserdem benötigt der Agent xinetd zum Ausliefern der Monitoring Daten.

Per SSH auf dem Server. (Auch hier: Name des .deb-Files ggf. anpassen)

```
sudo gdebi check-mk-agent_1.2.8p1-1_all.deb
```

Anschließend noch das Supernode-Plugin hinzufügen:

```
cd /usr/lib/check_mk_agent/local
sudo wget -O supernode https://raw.githubusercontent.com/eulenfunk/check_mk/master/supernode
sudo chmod 755 supernode
sudo chmod +x supernode
```

Der Rechner hält ab nun Daten zum Abruf bereit.

JJX Bescheid sagen, der kümmert sich dann darum.

Danach den Supernode rebooten.

Hier eine grafische Übersicht über die beteiligten Konfigurationsdateien auf dem Supernode:

Auf dem Konzentrator

Auf dem Konzentrator muss die zum Supernode passende Konfiguration angelegt werden:

```
cd /opt/eulenfunk/konzentrator/config
sudo nano meinestadt-1
```

Dort müssen folgende Werte eingetragen werden:

```
# Beschreibender Name "stadt-N"
SUPERNODE_NAME=meinestadt-1

# Soll die Netzwerkkonfiguration automatisch beim Systemstart gesetzt werden
AUTOSTART=1

# IPv4 Konfiguration
SUPERNODE_CLIENT_IPV4_NET=<IPv4 Netz fuer die Clients, 172.XX.0.0/16>
SUPERNODE_TRANS_IPV4_NET=<IPv4 Transit-Netz, 172.31.YYY.0/24>
SUPERNODE_TRANS_IPV4_REMOTE=<Supernode IPv4 eth1 Adresse Transit-Netz, 172.31.YYY.1>

# IPv6 Konfiguration
SUPERNODE_CLIENT_IPV6_NET=<IPv6 Netz fuer die Clients, 2a03:2260:AAAA:BBBB::/64>
SUPERNODE_TRANS_IPV6_NET=<IPv6 Supernetz fuer Transit, 2a03:2260:AAAA:BBBB::/56>
SUPERNODE_TRANS_IPV6_LOCAL=<IPv6 Supernetz lokale Adresse, 2a03:2260:AAAA:BBBB::1>
SUPERNODE_TRANS_IPV6_REMOTE=<IPv6 Supernetz remote Adresse, 2a03:2260:AAAA:BBB::2>
```

Man kann dann die Konfiguration folgendermaßen aktivieren:

```
cd /opt/eulenfunk/konzentrator
sudo ./supernode.sh start meinestadt-1
```

Die Konfiguration kann im laufenden Betrieb auch wieder entfernt werden (damit wird die Stadt allerdings vom Freifunk getrennt!)

```
cd /opt/eulenfunk/konzentrator
sudo ./supernode.sh stop meinestadt-1
```

Durch den Parameter AUTOSTART=1 wird beim Reboot des Konzentrators die Konfiguration für diese Stadt automatisch wieder gesetzt.

Den Konzentrator und den Supernode rebooten, um die Reboot-Festigkeit zu testen.

5.5 Nützliches

5.5.1 Benutzerkennwort zurücksetzen

Hat man sein Kennwort vergessen, kann man einen anderen Nutzer mit Źugriff auf den Server bitten ein neues Kennwort zu setzen

```
sudo passwd Meinbenutzername
```

CHAPTER 6

Map

Nodegraph und Geo-Karte sind unter

http://map.eulenfunk.de (für alle communities) verfügbar.

-tbd-