# Elbrus Analytics - Bereitstellungshandbuch

Tobias Schmidt

July 20, 2022

# 1  Server Infrastruktur

## 1.1  Initiale Server Konfiguration

Listing 1: Installieren des DNF Konfigurations Managers, um in Folge das Remote Docker Repository hinzuzufügen.

```
elbrus@server:~$ dnf install dnf-plugin-config-manager
```

Listing 2: Installieren und aktivieren von Docker.

```
elbrus@server:~$ sudo dnf config-manager \
 --add-repo https://download.docker.com/linux/centos/docker-ce.repo

elbrus@server:~$ sudo dnf install docker-ce docker-ce-cli containerd.io

elbrus@server:~$ systemctl enable docker

elbrus@server:~$ systemctl start docker
```

Listing 3: Berechtigt den User Elbrus 'sudo' zu verwenden. Berechtigt den User Elbrus darüber hinaus Docker ohne 'sudo' aufzurufen. Zudem wird dem User ein Heimverzeichnis angelegt, sowie die 'Bash' als standard Konsole gesetzt.

```
elbrus@server:~$ useradd -s /bin/bash -G docker,wheel -m elbrus

elbrus@server:~$ passwd elbrus
Changing password for user elbrus.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
elbrus@server:~$
```

Listing 4: Setzen der Zeitzone auf 'Europa/Wien'.

```
elbrus@server:~$ sudo timedatectl set-timezone Europe/Vienna
```

## 1.2  python

## 1.3  rust

# 2 Datenbank

# 3 Aufzeichnen der Daten

Listing 5: Installieren von 'tcpdump' für das aufzeichnen von Daten.

```
elbrus@server:~$ sudo dnf install tcpdump
```

Listing 6: Anlegen eines Users der Berechtigungen zum ausführen von 'tcpdump' erhält.

```
elbrus@server:~$ sudo useradd aragog
```

Listing 7: Zuweisen von 'tcpdump' zu der Gruppe 'aragog'.

```
elbrus@server:~$ sudo chgrp aragog /usr/sbin/tcpdump
```

Listing 8: Ändern der Berechtigungen auf 'tcpdump'.

```
elbrus@server:~$ chmod 750 /usr/sbin/tcpdump
elbrus@server:~$ sudo setcap cap_net_raw,cap_net_admin=eip \
/usr/sbin/tcpdump
```

Kopieren von '.env.example', 'capture.service.example', 'elb-capture.sh', 'elb-capture-log.sh', 'elb-capture-postrotate.sh' in beliebigen Ordner

Listing 9: Anhand von '.env.example' eigene '.env' Datei anlegen

```
1   #where the log should be stored
2   LOGFILE="/var/elbrus/capture/capture.log"
3   #where the traces should be stored
4   PCAP="/var/elbrus/capture/pcap/"
5   #how much time each trace should contain in seconds
6   TIMEPERCAPTURE=900
7   #the maximum amount of files
8   MAXFILES=10
9   #the interface to capture on
10  INTERFACE=eth0
11  #the path to the 'elb-capture-postrotate.sh' script
12  POSTROTATEPATH=/var/elbrus/capture/elb-capture-postrotate.sh
```

Listing 10: capture.service.example - Die Variable 'WorkingDirectory', Die Variable 'User' sowie die Variable 'ExecStopPost' anpassen.

```
3   ...
4   #job is starting immediatly after the start action has been
    called
```

```
 5      Type=simple
 6      #the user to execute the script
 7      User=aragog
 8      #the working directory
 9      WorkingDirectory=/var/elbrus/capture
10      #which script should be executed
11      ExecStart=/bin/bash elb-capture.sh
12      #when the script should restart
13      Restart=on-failure
14      #set the restart timeout
15      RestartSec=5
16      #which script should be executed when the service stops
17      ExecStopPost=/bin/bash elb-capture-log.sh
18
19      [Install]
20      ...
```

Listing 11: Kopieren des Serviceprogrammes

```
elbrus@server:~$ cp capture.service.example \
/etc/systemd/system/capture.service
```

Listing 12: Neuladen des 'systemctl' Deamons

```
elbrus@server:~$ systemctl daemon-reload
```

Listing 13: Aktivieren des Serviceprogrammes

```
elbrus@server:~$ systemctl enable capture.service
```

Listing 14: Starten des Serviceprogrammes

```
elbrus@server:~$ systemctl start capture.service
```

# 4 Packet Capture Importer

# 5  Report Generator

# 6 SNMP Manager

# 7 SSH Manager

## 7.1 Umgebung Konfigurieren

Kopieren von 'requirements.txt', '.env.example', 'initialise.sh', 'routine.sh', 'setup.sh', 'main.py', 'ssh-manager.service.example', 'ssh-manager-schedule.timer.example' in den selben beliebigen Ordner

Listing 15: Anhand von '.env.example' eigene '.env' Datei anlegen

```
1   #values regarding the jumpserver:
2   #IP, PORT and USER values must be set!
3   #depending on the usage you can set either:
4   #    -PASS and KEYFILE: the keyfile is used, the pass is
    interpreted as the passphrase
5   #    -only KEYFILE: the keyfile is used
6   #    -only PASS: the password is used as is regular credentials
7   JUMPSERVER_IP="2.2.2.15"
8   JUMPSERVER_PORT=22
9   JUMPSERVER_USER=admin
10  JUMPSERVER_PASS=password
11  SSH_KEYFILE='my/sample/path'
12
13  #all database values must be set!
14  POSTGRES_HOST="192.168.0.1"
15  POSTGRES_PORT=245
16  POSTGRES_DB=mydb
17  POSTGRES_USER=admin
18  POSTGRES_PASS=password
19
20  #paths are configured by running 'setup.sh'
21  CONFIGPATH="/thats/where/i/store/my/configs"
22  MAINPATH="/the/path/to/main.py"
```

### 7.1.1   1 - Mit Setup script

Listing 16: Ausführen des setup Scripts

```
elbrus@server:~$ cd ssh-manager/src
elbrus@server:~/ssh-manager/src$ ./setup.sh
Setup for ssh-manager
Do you want to proceed? (y/n) y
we will proceed

Where do you want the config to be stored: (abolut path) /my/sample/path
Where is the 'main.py' file stored: (abolut path) /path/to/main.py

Do you want to store the config files at "/my/sample/path"?
Is your 'main.py' stored at "/path/to/main" (y/n/exit) y

The paths have been set!

Do you want to configure the systemd Service? (y/n/exit) y

Which User should execute the Service? elbrus

The systemd Service has been configured!

Do you want to run the initialise script? (y/n/exit) y

...

finished setup

elbrus@server:~/ssh-manager/src$
```

### 7.1.2   2 - Ohne Setup script

Listing 17: '.env' - Datei anpassen

```
17      ...
18      POSTGRES_PASS=password
19
20      #paths are configured by running 'setup.sh'
21      CONFIGPATH="/thats/where/i/store/my/configs"
22      MAINPATH="/the/path/to/main.py"
```

Listing 18: ssh-manager.service.example - Die Variable 'WorkingDirectory' sowie die Variable 'User' anpassen.

```
5       ...
6       #job is starting immediatly after the start action has been
        called
7       Type=simple
8       #the user to execute the script
9       User=elbrus
10      #the working directory
11      WorkingDirectory=/home/elbrus/Desktop/ssh-manager/src/
12      #which script should be executed
13      ExecStart=/bin/bash routine.sh
14      ...
```

Listing 19: Ausführen des Scripts zur Initialisierung des VCS Verzeichnisses.

```
elbrus@server:~$ ssh-manager/src/initialise.sh
```

## 7.2   Abhängigkeiten

Listing 20: Installieren von fehlenden python3 Packages.

```
elbrus@server:~$ pip3 install -r ssh-manager/requirements.txt
```

## 7.3  Automatisches ausführen des Skripts

Listing 21: Kopieren des Serviceprogrammes

```
elbrus@server:~$ cp src/ssh-manager.service.example \
 /etc/systemd/system/ssh-manager.service
```

Listing 22: Kopieren des Zeitplanungsprogrammes.

```
elbrus@server:~$ cp src/ssh-manager-schedule.timer.example \
 /etc/systemd/system/ssh-manager-schedule.timer
```

Listing 23: Neuladen des 'systemctl' Deamons

```
elbrus@server:~$ systemctl daemon-reload
```

Listing 24: Aktivieren des Serviceprogrammes

```
elbrus@server:~$ systemctl enable ssh-manager.service
```

Listing 25: Aktivieren des Zeitplanungsprogrammes

```
elbrus@server:~$ systemctl enable ssh-manager-schedule.timer
```

Listing 26: Aktivieren des Zeitplanungsprogrammes

```
elbrus@server:~$ systemctl enable ssh-manager-schedule.timer
```

Listing 27: Starten des Zeitplanungsprogrammes

```
elbrus@server:~$ systemctl start ssh-manager-schedule.timer
```

# 8 API

# 9 Webinterface