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

```
#where the log should be stored
LOGFILE="/var/elbrus/capture/capture.log"
#where the traces should be stored
PCAP="/var/elbrus/capture/pcap/"
#how much time each trace should contain in seconds
TIMEPERCAPTURE=900
#the maximum amount of files
MAXFILES=10
#the interface to capture on
INTERFACE=eth0
#the path to the ’elb-capture-postrotate.sh’ script
POSTROTATEPATH=/var/elbrus/capture/elb-capture-postrotate.sh
```

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

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

```
Type=simple
#the user to execute the script
User=aragog
#the working directory
WorkingDirectory=/var/elbrus/capture
#which script should be executed
ExecStart=/bin/bash elb-capture.sh
#when the script should restart
Restart=on-failure
#set the restart timeout
RestartSec=5
#which script should be executed when the service stops
ExecStopPost=/bin/bash elb-capture-log.sh

[Install]
...
```

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 beliebigen Ordner

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

```
#values regarding the jumpserver:
#IP, PORT and USER values must be set!
#depending on the usage you can set either:
#   -PASS and KEYFILE: the keyfile is used, the pass is interpreted as the passphrase
#   -only KEYFILE: the keyfile is used
#   -only PASS: the password is used as is regular credentials
JUMPSERVER_IP="2.2.2.15"
JUMPSERVER_PORT=22
JUMPSERVER_USER=admin
JUMPSERVER_PASS=password
SSH_KEYFILE='my/sample/path'

#all database values must be set!
POSTGRES_HOST="192.168.0.1"
POSTGRES_PORT=245
POSTGRES_DB=mydb
POSTGRES_USER=admin
POSTGRES_PASS=password
```

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: /my/sample/path

Do you want to store the config files at "/my/sample/path"? (y/n/exit) y

The path has been set to "/my/sample/path"!

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:
1. /my/sample/path
2. /ssh-manager/src
elbrus@server:~/ssh-manager/src$
```

7.1.2 2 - Ohne Setup script

Listing 17: initialise.sh - Die Variable ’DIR’

```
...

#directory in which the config is stored
DIR="/home/elbrus/Desktop/ssh-manager/config"

if [ -d "$DIR" ]; then
...
```

Listing 18: routine.sh - Die Variable ’DIR’, Den Pfad zum python Script

```
...

#directory in which the config is stored
DIR="/home/elbrus/Desktop/ssh-manager/config"

echo "info: retrieving configurations"
#execute python job
python3 /home/elbrus/Desktop/ssh-manager/src/main.py

#set current date in the Format YYYY-MM-DD-HH:MM:SS
...
```

Listing 19: main.py - Die Variable ’directory’

```
...

#directory in which the output is stored
directory = ’/home/elbrus/Desktop/ssh-manager/config/’
#address of current endpoint
address = None
...
```

Listing 20: ssh-manager.service.example - Die Variable ’WorkingDirectory’, Die Variable ’User’

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

```
Type=simple
#the user to execute the script
User=elbrus
#the working directory
WorkingDirectory=/home/elbrus/Desktop/ssh-manager/src/
#which script should be executed
ExecStart=/bin/bash routine.sh
...
```

Listing 21: Ausführen des Scripts zur Initialisierung des VCS Verzeichnisses.

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

7.2 Abhängigkeiten

Listing 22: Installieren von fehlenden python3 Packages.

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

7.3 Automatisches ausführen des Skripts

- In 'ssh-manager.service.example' den Pfad des Arbeitsverzeichnisses ändern
- Den Benutzer in 'ssh-manager.service.example' ändern

Listing 23: Kopieren des Serviceprogrammes

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

Listing 24: Kopieren des Zeitplanungsprogrammes.

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

Listing 25: Neuladen des 'systemctl' Deamons

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

Listing 26: Aktivieren des Serviceprogrammes

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

Listing 27: Aktivieren des Zeitplanungsprogrammes

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

Listing 28: Aktivieren des Zeitplanungsprogrammes

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

Listing 29: Starten des Zeitplanungsprogrammes

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

8 API

9 Webinterface