Elbrus Analytics - Bereitstellungshandbuch

Tobias Schmidt

July 21, 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

2.1 Voraussetzungen

Listing 5: Hinzufügen des PostgreSQL Drittanbieter-Repository, um die neuesten PostgreSQL-Pakete zu erhalten.

```
elbrus@server:~$ sudo yum install \
 https://download.postgresql.org/pub/repos/yum/reporpms/ \
 EL-$(rpm -E %{rhel})-x86_64/pgdg-redhat-repo-latest.noarch.rpm
```

Listing 6: Erstellen des Timescale repository.

```
elbrus@server:~$ sudo tee /etc/yum.repos.d/timescale_timescaledb.repo
```

Listing 7: Berarbeiten des Timescale repository.

```
1  [timescale_timescaledb]
2  name=timescale_timescaledb
3  baseurl=https://packagecloud.io/timescale/timescaledb/el/$(rpm
   -E %{rhel})/\$basearch
4  repo_gpgcheck=1
5  gpgcheck=0
6  enabled=1
7  gpgkey=https://packagecloud.io/timescale/timescaledb/gpgkey
8  sslverify=1
9  sslcacert=/etc/pki/tls/certs/ca-bundle.crt
10 metadata_expire=300
```

Listing 8: Updaten der lokalen Package-Liste.

```
elbrus@server:~$ sudo yum update
```

Listing 9: Installieren von TimescaleDB

```
elbrus@server:~$ sudo dnf -qy module disable postgresql
elbrus@server:~$ sudo dnf install postgresql14 postgresql14-server -y
elbrus@server:~$ sudo dnf install timescaledb-2-postgresql-14 -y
```

2.2 Umgebung Konfigurieren

Listing 10: Initialisieren der Datenbank.

```
elbrus@server:~$ /usr/pgsql-14/bin/postgresql-14-setup initdb
```

Listing 11: Verknüpfen von 'postgresql' Serive Start mit Serverstart sowie den Service starten.

```
elbrus@server:~$ sudo systemctl enable postgresql-14
elbrus@server:~$ sudo systemctl start postgresql-14
```

Listing 12: var/lib/pgsql/14/data/postgresql.conf - Ändern der folgenden Zeilen

```
- #shared_preload_libraries = ''
+ shared_preload_libraries = 'timescaledb'
```

Listing 13: Anpassen der Datenbank Einstellungen auf die Server Hardware.

```
elbrus@server:~$ sudo timescaledb-tune --pg-config=/usr/ \
 pgsql-14/bin/pg_config --yes
```

Listing 14: Neustarten des Services um Änderungen zu übernehmen.

```
elbrus@server:~$ sudo systemctl restart postgresql-14
```

2.3 Erstellen der Elbrus-Datenbank

Listing 15: Verbinden mit dem interaktiven Terminal von 'postgres'.

```
elbrus@server:~$ sudo su postgres -c psql
```

Im folgenden Text sind markierte Abschnitte Variablen, welche im darunterliegen SQL gerändert werden können, was aus Sichertsgründen dringend empfohlen wird.

1. Die Datenbank elbrus anlegen
2. Die Zeitzone auf Europe/Vienna setzen
3. Den User elbrus mit dem Passwort elbrus123! anlegen
4. Dem User alle rechte auf die voher erstellte Datenbank geben

Listing 16: Aufführen von SQL Befehlen.

```
CREATE DATABASE elbrus;
ALTER DATABASE elbrus SET timezone TO 'Europe/Vienna';
CREATE USER elbrus PASSWORD 'elbrus123!';
GRANT ALL ON DATABASE elbrus TO elbrus;
```

Listing 17: Wechseln zu erstellter Datenbank

```
\c elbrus
```

Listing 18: Hinzufügen der TimescaleDB Erweiterung.

```
CREATE EXTENSION IF NOT EXISTS timescaledb;
exit
```

Kopieren von 'init.sql' in beliebigen Ordner.

Listing 19: Anlegen der benötigten Tabellen duch das ausführen von 'init.sql'.

```
elbrus@server:~$ psql -U elbrus -d elbrus -f init.sql
```

3 Aufzeichnen der Daten

3.1 Voraussetzungen

Listing 20: Installieren von 'tcpdump' für das aufzeichnen von Daten.

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

Listing 21: Anlegen eines Users der Berechtigungen zum ausführen von 'tcpdump' erhält.

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

Listing 22: Zuweisen von 'tcpdump' zu der Gruppe 'aragog'.

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

Listing 23: 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
```

3.2 Umgebung Konfigurieren

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

Listing 24: Anhand von '.env.example' eigene '.env' Datei anlegen

```
1  #where the log should be stored
2  LOGFILE=/var/elbrus/capture/capture"-$(date +"%Y-%U")".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 POSTROTATESCRIPT=/var/elbrus/capture/elb-capture-postrotate.sh
```

3.3 Der Systemd Service

Listing 25: 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 26: Kopieren des Serviceprogrammes

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

Listing 27: Neuladen des 'systemctl' Deamons

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

Listing 28: Aktivieren des Serviceprogrammes

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

Listing 29: 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 30: 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: keyfile is used with 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

    #paths are configured by running 'setup.sh'
    CONFIGPATH="/thats/where/i/store/my/configs"
    MAINPATH="/the/path/to/main.py"
```

7.1.1 1 - Mit Setup script

Listing 31: 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 32: .env - Die Variable 'CONFIGPATH' sowie die Variable 'MAINPATH' 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 33: 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 34: Ausführen des Scripts zur Initialisierung des VCS Verzeichnisses.

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

7.2 Voraussetzungen

Listing 35: Installieren von fehlenden python3 Packages.

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

7.3 Der Systemd Service

Listing 36: Kopieren des Serviceprogrammes

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

Listing 37: Kopieren des Zeitplanungsprogrammes.

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

Listing 38: Neuladen des 'systemctl' Deamons

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

Listing 39: Aktivieren des Serviceprogrammes

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

Listing 40: Aktivieren des Zeitplanungsprogrammes

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

Listing 41: Aktivieren des Zeitplanungsprogrammes

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

Listing 42: Starten des Zeitplanungsprogrammes

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

8 API

9 Webinterface