#Configuration of secure InfluxV2 Server with mosquitto mqtt

#Basics: Each plant get its own VM

#VM has fixed IP or dyndns Service and domain DNS setting for machine already setup

#Open datacenter firewall for ports: 80, 443, 8883

#Used parameters, recommended to predefine before deployment

#<hostname> - planttype\_operator\_location (demo) equals all user names

#<domain> - demo.sms-metrics.com

#<password\_ubuntu\_user> (sudo root)

#<password\_sshkey>

#<password\_mqtt>

#<password\_influx>

#<admin\_e-mail>

#create VM, ssh into machine with given credentials by cloud supplier

#use key instead of passwords for SSH if possible and deactivate password login

#create ssh key on remote machine:

ssh-keygen -t rsa

#use <hostname> as certificate name and <password\_sshkey> as key password

ssh-copy-id -i /home/user/.ssh/<hostname> <hostname>at<domain>

#login via ssh

ssh -i /home/user/.ssh/<hostname> <hostname>at<domain>

#use <password\_sshkey> to login

#update ubuntu

sudo apt update & sudo apt upgrade

#add user <domain> as he is used to run the services

sudo adduser <domain>

#ssh settings

vi /etc/ssh/sshd\_config

#deactivate password based login

#as SSH client termius.com is recommended. When started look for the hidden button to continue without account (it works also without)

#unattended upgrades einrichten:

# Folgende Datei mit nano bearbeiten: /etc/apt/apt.conf.d/50unattended-upgrades

# Folgende Zeilen entkommentieren bzw. anpassen:

"${distro\_id}:${distro\_codename}-updates";

"${distro\_id}:${distro\_codename}-proposed";

Unattended-Upgrade::Automatic-Reboot "true"; #idealerweise nachts!

#restart machine und relogin (with certificate)

sudo reboot now

ssh -i /home/user/.ssh/<hostname> <hostname>at<domain>

#Firewall settings usually set through cloud provider interface, otherwise use ufw.

#certbot certificate:

#make sure nothing installed that binds port 80, uninstall apache or nginx

sudo add-apt-repository ppa:certbot/certbot

sudo apt install certbot

sudo certbot certonly --standalone --preferred-challenges http -d <domain>

#follow instructions, <admin\_e-mail> required

#mosquitto mqtt broker:

sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa

sudo apt update

sudo apt install mosquitto mosquitto-clients

sudo mosquitto\_passwd -c /etc/mosquitto/passwd <hostname>

# will be prompted twice for <password\_mqtt>

sudo vi /etc/mosquitto/conf.d/default.conf

#add and safe following content; replace <domain>

#----------------------start

allow\_anonymous false

password\_file /etc/mosquitto/passwd

listener 1883 localhost

listener 8883

certfile /etc/letsencrypt/live/<domain>/cert.pem

cafile /etc/letsencrypt/live/<domain>/chain.pem

keyfile /etc/letsencrypt/live/<domain>/privkey.pem

#----------------------end

#install telegraf

sudo -s

sudo curl -sL https://repos.influxdata.com/influxdb.key | apt-key add -

source /etc/lsb-release

echo "deb https://repos.influxdata.com/${DISTRIB\_ID,,} ${DISTRIB\_CODENAME} stable" | tee -a /etc/apt/sources.list

sudo apt update

exit

sudo apt install telegraf

cd /etc/telegraf/

sudo mv telegraf.conf telegraf.old

sudo vi telegraf.conf

#--------start

# Configuration for telegraf agent

[agent]

interval = "1s"

round\_interval = true

metric\_batch\_size = 1000

metric\_buffer\_limit = 10000

collection\_jitter = "0s"

flush\_interval = "1s"

flush\_jitter = "0s"

precision = "ms"

debug = false

quiet = false

logfile = ""

hostname = ""

omit\_hostname = true

# Configuration for sending metrics to InfluxDB

[[outputs.influxdb]]

urls = ["http://127.0.0.1:8086"]

database = "metrics"

retention\_policy = "12w"

# # Read metrics from MQTT topic(s), json

[[inputs.mqtt\_consumer]]

name\_override = "json"

servers = ["tcp://localhost:1883"]

qos = 2

connection\_timeout = "30s"

max\_undelivered\_messages = 1000

topics = [

"json/#",

]

persistent\_session = false

username = "<hostname>"

password = "<password\_mqtt>"

data\_format = "json"

json\_query = ""

json\_name\_key = "json"

json\_time\_key = ""

json\_time\_format = ""

#---------stop

#install influx v2 (no repository yet available, hence manual setup required)

cd /home/<domain>/

wget https://dl.influxdata.com/influxdb/releases/influxdb\_2.0.0-beta.12\_linux\_amd64.tar.gz

tar xvfz influxdb\_2.0.0-beta.12\_linux\_amd64.tar.gz

# move executables into proper folder:

sudo cp influxdb\_2.0.0-alpha.14\_linux\_amd64/{influx,influxd} /usr/local/bin/

#change to folder where all services are organized (service= software that is started at boot time and restarted in case it crashes)

#see also: https://devconnected.com/how-to-install-influxdb-on-ubuntu-debian-in-2019/

#start influxd manually

# use SSH port Forwarding to connect to port :9999 and fill in the get started information

# use <domain> as username and <password\_influx> as password

cd /lib/systemd/system

sudo vi influxdb2.service

#--------------------start

[Unit]

Description=InfluxDB 2.0 service file.

Documentation=https://v2.docs.influxdata.com/v2.0/get-started/

After=network-online.target

[Service]

User=<domain>

Group=<domain>

ExecStart=/usr/local/bin/influxd \

--tls-cert=/etc/letsencrypt/live/<domain>.sms-metrics.com/cert.pem \

--tls-key=/etc/letsencrypt/live/<domain>.sms-metrics.com/privkey.pem \

--http-bind-address=:443 \

--vault-tls-server-name=<domain>.sms-metrics.com

Restart=on-failure

[Install]

WantedBy=multi-user.target

#------------------------stop

#influx needs to get the right to access port 443:

sudo -s

setcap 'cap\_net\_bind\_service=+ep' /usr/local/bin/influxd

logout

#influx needs to access the certificates from letsencrypt:

sudo chmod -R 777 /etc/letsencrypt

# activate service

sudo systemctl enable influxdb2.service

# ToDo:

# setcap must be executed after each update

# access to binary and certificates is not limited (setup should be changed to be more restrictive to increase security)

#fail2ban brute force protection

sudo apt install fail2ban

sudo vi /etc/f ail2ban/jail.conf

#create following entry

#---------start

[mosquitto]

enabled = true

filter = mosquitto

action = iptables[name=mosquitto, port=8883, protocol=tcp]

logpath = /var/log/mosquitto/mosquitto.log

maxretry = 2

bantime = 6000 sec

#----------end

#create filter

sudo vi /etc/fail2ban/filter.d/mosquitto.conf

#add following text and save

#------------ start

[Init]

maxlines = 2

[Definition]

failregex = .+ New connection from <HOST> on port \d+.\n.+Socket error on client <unknown>

ignoreregex =

#--------------end

#cron job to restart influx and mosquitto to take over new certificates

sudo crontab -e

#add lines:

0 3 \* \* 1 systemctl stop mosquitto

5 3 \* \* 1 systemctl start mosquitto

10 3 \* \* 1 systemctl restart influxv2.service

#restart

sudo reboot now

#Good Luck

#debug information

telegraf --config /etc/telegraf/telegraf.conf

vi /var/log/mosquitto

vi /var/log/fail2ban