nano /etc/ssh/sshd\_config – открываем файл sshd\_config

systemctl restart sshd – перезапуск службы

Создаем файл с конфигурацией. Отключаем все лишнее (графическую оболочку, разные службы и т.д.), освобождаем лишние ресурсы. EOF – это окончание файла. GRUB\_TIMEOUT=0 – убираем таймаут при загрузке системы.

cat > /etc/default/grub <<EOF

GRUB\_DEFAULT=0

GRUB\_TIMEOUT=0 – убираем таймаут при загрузке системы

GRUB\_DISTRIBUTOR=`lsb\_release -i -s 2> /dev/null || echo Debian`

GRUB\_CMDLINE\_LINUX\_DEFAULT="quiet text mitigations=off nowatchdog processor.ignore\_ppc=1 cpufreq.default\_governor=performance ipv6.disable=1 apparmor=0 selinux=0 debug=-1"

GRUB\_CMDLINE\_LINUX=""

GRUB\_DISABLE\_LINUX\_RECOVERY=true

GRUB\_DISABLE\_OS\_PROBER=true

GRUB\_TERMINAL=console

EOF

Команда обновит загрузчик и перезапустит систему

update-grub && \

reboot

В случае если указанная выше команда не запускается, попробовать запустить export PATH=$PATH:/usr/sbin

Далее запускаем

echo "net.ipv4.ip\_forward=1" >> /etc/sysctl.conf && \

echo "net.ipv4.conf.all.forwarding=1" >> /etc/sysctl.conf && \

sysctl -p /etc/sysctl.conf && \

systemctl stop cron && \

systemctl stop apparmor && \

systemctl stop console-setup && \

systemctl stop keyboard-setup && \

systemctl disable cron && \

systemctl disable apparmor && \

systemctl disable console-setup && \

systemctl disable keyboard-setup && \

systemctl set-default multi-user.target && \

apt install -y sudo curl wget gnupg2 systemd-timesyncd htop && \

sed -i 's/#NTP=/NTP=1.ru.pool.ntp.org/' /etc/systemd/timesyncd.conf && \

systemctl restart systemd-timesyncd

Если возникает ошибка bash: sysctl: command not found, то запускаем команду export PATH="/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin" и повторяем запрос выше.

df –h – проверим сколько места на диске занимает система

htop – проверим сколько ресурсов потребляет система

Установка ПО: Mosquitto, Node-RED, WireGuard, Influx, Telegraf, Grafana

После перезагрузки снова коннектимся по SSH:

ssh root@192.168.50.56

***Установка Mosquitto***

Добавляем пакет и проверяем актуальную версию, должна быть 2.0.18:

sudo wget -qO- https://repo.mosquitto.org/debian/mosquitto-repo.gpg.key | gpg --dearmor -o /etc/apt/keyrings/mosquitto-repo.gpg && \  
sudo wget -O /etc/apt/sources.list.d/mosquitto-bookworm.list https://repo.mosquitto.org/debian/mosquitto-bookworm.list && \  
sudo apt update && \  
sudo apt-cache show mosquitto | grep Version

Устанавливаем:

sudo apt install -y mosquitto=2.0.18-0mosquitto1~bookworm1 mosquitto-clients=2.0.18-0mosquitto1~bookworm1 && \

Создаем пароль для учетной записи IoT (после ввода команды нужно дважды ввести пароль) и настраиваем права доступа:

sudo mosquitto\_passwd -c /etc/mosquitto/passwd IoT && \  
sudo chmod 777 /etc/mosquitto/passwd

Создаем конфигурацию:

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

И вставляем туда следующие содержимое:

allow\_anonymous **false**  
password\_file /**etc**/**mosquitto**/**passwd**  
**listener** 1883

Записываем файл ctrl+o, enter и выходим из редактора ctrl+x.

Сохраненная только что конфигурация запрещает анонимное подключение, без связки логин-пароль, указывает путь к файлу с заданным ранее паролем для пользователя IoT, а также задает порт, на котором будет работать MQTT брокер. Перезапускаем службу:

systemctl restart mosquitto

mosquitto\_sub -h 192.168.1.114 -p 1883 -t GB -u "IoT" -P "8" - ввести команду

Если возникает ошибка bash: mosquitto\_sub: command not found, ввести apt-get install mosquitto-clients.

Далее отправляем сообщение с помощью команды mosquitto\_pub -h 192.168.1.114 -p 1883 -t "GB" -m "Hello, GB!" -u "IoT" -P "8".

Устанавливаем Node.js 20 + node-red + админка

sudo apt-get update && sudo apt-get install -y ca-certificates curl gnupg && \

curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg && \

NODE\_MAJOR=22 && \

echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node\_$NODE\_MAJOR.x nodistro main" | sudo tee /etc/apt/sources.list.d/nodesource.list && \

sudo apt-get update && sudo apt-get install nodejs -y && \

bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered) && \

sudo systemctl enable nodered && \

sudo systemctl start nodered

node-red – запускаем службу

# Установка WireGuard

sudo curl -O https://raw.githubusercontent.com/angristan/wireguard-install/master/wireguard-install.sh && \

sudo chmod +x wireguard-install.sh && \

sudo ./wireguard-install.sh

sudo nano /etc/wireguard/wg0.conf

'remove all ipv6'

sudo systemctl restart wg-quick@wg0

reboot

Установим приложение WireGuard и запустим его.

Командой wg show посмотрим статистику обмена.

# Установка InfluxDB2 + Telegraf + Grafana (версии надо актуализировать вручную на сайтах по загрузке)

sudo wget -q https://repos.influxdata.com/influxdata-archive\_compat.key && \

sudo echo '393e8779c89ac8d958f81f942f9ad7fb82a25e133faddaf92e15b16e6ac9ce4c influxdata-archive\_compat.key' | sha256sum -c && cat influxdata-archive\_compat.key | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/influxdata-archive\_compat.gpg > /dev/null && \

sudo echo 'deb [signed-by=/etc/apt/trusted.gpg.d/influxdata-archive\_compat.gpg] https://repos.influxdata.com/debian stable main' | sudo tee /etc/apt/sources.list.d/influxdata.list && \

sudo apt-get install -y adduser libfontconfig1 musl && \

sudo wget https://dl.grafana.com/oss/release/grafana\_10.4.3\_amd64.deb && \

sudo dpkg -i grafana\_10.4.3\_amd64.deb && \

sudo rm /root/grafana\_10.4.3\_amd64.deb && \

sudo systemctl enable grafana-server && \

sudo systemctl start grafana-server && \

sudo apt-get update && sudo apt-get install -y influxdb2 telegraf && \

sudo systemctl start influxd && \

sudo systemctl enable telegraf

sudo cat > /etc/telegraf/telegraf.conf <<EOF

# Configuration for telegraf agent

[agent]

interval = "15s"

round\_interval = true

metric\_batch\_size = 250

metric\_buffer\_limit = 2500

collection\_jitter = "0s"

flush\_interval = "15s"

flush\_jitter = "0s"

precision = ""

hostname = ""

omit\_hostname = false

[[outputs.influxdb\_v2]]

urls = ["http:// 192.168.171.129:8086"]

token = "LpdCJ55vm4lExGg1WZVwOXXuTALVA-oQ82PwvzKTRW2Y62\_1iKJ0peiT0eEsTC5YBqPOkggGRiaXpzhHmYl51w=="

organization = "IoT"

bucket = "IoT"

[[inputs.mqtt\_consumer]]

servers = ["tcp:// 192.168.171.129:1883"]

topics = ["#"]

username = "IoT"

password = "8"

data\_format = "value"

data\_type = "float"

EOF

mosquitto\_pub -h 192.168.1.114 -t "GB" -m "28.5" -u "IoT" -P "8"