Instalace RPI monitoru

https://rpi-experiences.blogspot.com/

- Instalace RPI-monitoru

Instalace RPI-monitoru

Podrobný návod na instalaci RPi-Monitor s funk?ním zobrazením hodnot na RPi 5

1. Vytvo?ení adresá?ové struktury

```
sudo mkdir -p /opt/rpi-monitor/data
sudo mkdir -p /opt/rpi-monitor/conf
sudo chmod -R 777 /opt/rpi-monitor
```

2. Vytvo?ení konfigura?ních soubor?

Vytvoření system.conf

```
sudo nano /opt/rpi-monitor/conf/system.conf
```

Zkopíruj a vlož následující obsah do otevřeného souboru:

```
# RPi-Monitor configuration for RPi 5
# This file should be saved to /opt/rpi-monitor/conf/system.conf

################################################################
# Configuration for temperature monitoring and CPU frequency
################################################################

# Define temperature monitoring
dynamic.1.name=temperature
dynamic.1.source=/sys/class/thermal/thermal_zone0/temp
dynamic.1.regexp=(.*)
dynamic.1.postprocess=$1/1000
dynamic.1.rrd=GAUGE

# Define CPU frequency (compatible with RPi 5)
dynamic.2.name=cpu_frequency
dynamic.2.source=cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
dynamic.2.regexp=(.*)
dynamic.2.postprocess=$1/1000
```

```
dynamic.2.rrd=GAUGE

# Define scaling governor
dynamic.3.name=scaling_governor
dynamic.3.source=cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
dynamic.3.regexp=(.*)
dynamic.3.postprocess=
dynamic.3.rrd=

#################################################################
# Define Memory stats
#################################################################

# Define memory monitoring
dynamic.4.name=memory_usage
dynamic.4.source=free -b
dynamic.4.regexp=Mem: +([0-9]+) +([0-9]+) +([0-9]+) +([0-9]+) +([0-9]+) +([0-9]+)
dynamic.4.postprocess=
dynamic.4.rrd=GAUGE

# Define swap monitoring
dynamic.5.name=swap_usage
dynamic.5.source=free -b
dynamic.5.regexp=Swap: +([0-9]+) +([0-9]+) +([0-9]+)
dynamic.5.postprocess=
dynamic.5.rrd=GAUGE

#################################################################
# Define Uptime stats
#################################################################

# Define uptime
dynamic.6.name=uptime
dynamic.6.source=cat /proc/uptime
dynamic.6.regexp=^([0-9]*)\.([0-9]*)
dynamic.6.postprocess=
dynamic.6.rrd=GAUGE

#################################################################
# Define CPU utilization
```

```
#################################################################

dynamic.7.name=cpu_loading
dynamic.7.source=/proc/stat
dynamic.7.regexp=^cpu  +([0-9]+) ([0-9]+) ([0-9]+) ([0-9]+) ([0-9]+) ([0-9]+) ([0-9]+) ([0-
9]+) ([0-9]+) ([0-9]+)
dynamic.7.postprocess=
dynamic.7.rrd=DERIVE

#################################################################
# Define web interface appearance settings
#################################################################

web.status.1.content.1.name=Processor
web.status.1.content.1.icon=cpu.png
web.status.1.content.1.line.1="Temperature: <b>"+data.temperature+"&deg;C</b>"
web.status.1.content.1.line.2="CPU frequency: <b>"+data.cpu_frequency+"MHz</b>"
web.status.1.content.1.line.3="Governor: <b>"+data.scaling_governor+"</b>"

web.status.1.content.2.name=Memory
web.status.1.content.2.icon=memory.png
web.status.1.content.2.line.1="Used: <b>"+KMG(data.memory_usage[1],2)+"</b> Available:
<b>"+KMG(data.memory_usage[2],2)+"</b> Total: <b>"+KMG(data.memory_usage[0],2)+"</b>"
web.status.1.content.2.line.2="Swap: <b>"+KMG(data.swap_usage[1],2)+"</b> /
<b>"+KMG(data.swap_usage[0],2)+"</b>"

web.status.1.content.3.name=Uptime
web.status.1.content.3.icon=uptime.png
web.status.1.content.3.line.1="Uptime: <b>"+Uptime(data.uptime[0])+"</b>"

# CPU Load and usage graph
web.statistics.1.content.1.name=CPU Loading
web.statistics.1.content.1.graph.1=cpu_loading[0,1,2,3,4,5,6,7,8,9]
web.statistics.1.content.1.ds_graph_options.cpu_loading[0].label=User
web.statistics.1.content.1.ds_graph_options.cpu_loading[1].label=Nice
web.statistics.1.content.1.ds_graph_options.cpu_loading[2].label=System
web.statistics.1.content.1.ds_graph_options.cpu_loading[3].label=Idle
web.statistics.1.content.1.ds_graph_options.cpu_loading[4].label=Iowait
web.statistics.1.content.1.ds_graph_options.cpu_loading[5].label=IRQ
web.statistics.1.content.1.ds_graph_options.cpu_loading[6].label=SoftIRQ
```

```
web.statistics.1.content.1.ds_graph_options.cpu_loading[7].label=Steal
web.statistics.1.content.1.ds_graph_options.cpu_loading[8].label=Guest
web.statistics.1.content.1.ds_graph_options.cpu_loading[9].label=GuestNice

# Temperature Graph
web.statistics.1.content.2.name=Temperature
web.statistics.1.content.2.graph.1=temperature
```

Stiskni CTRL+X, poté Y a Enter pro uložení souboru.

Vytvo?ení system_info.conf

```
sudo nano /opt/rpi-monitor/conf/system_info.conf
```

Zkopíruj a vlož následující obsah:

```
# System info configuration for RPi 5
# Save this as /opt/rpi-monitor/conf/system_info.conf

##############################################################
# Define System info from custom script
##############################################################

dynamic.1000.name=system_info
dynamic.1000.source=/data/system_info.sh
dynamic.1000.regexp=processor_model=(.*)\ndistribution=(.*)\nkernel_version=(.*)\nfirmware=(.*
)\npackages=(.*)
dynamic.1000.postprocess=

##############################################################
# Define appearance
##############################################################

web.status.1.content.10.name=System
web.status.1.content.10.icon=system.png
web.status.1.content.10.line.1="Processor: <b>"+data.system_info[0]+"</b>"
web.status.1.content.10.line.2="Distribution: <b>"+data.system_info[1]+"</b>"
web.status.1.content.10.line.3="Kernel version: <b>"+data.system_info[2]+"</b>"
web.status.1.content.10.line.4="Firmware: <b>"+data.system_info[3]+"</b>"
web.status.1.content.10.line.5="Package(s): <b>"+data.system_info[4]+"</b>"
```

Stiskni CTRL+X, poté Y a Enter pro uložení souboru.

Vytvo?ení skriptu system_info.sh

```
sudo nano /opt/rpi-monitor/data/system_info.sh
```

Zkopíruj a vlož následující obsah:

```
#!/bin/bash
# Script to gather system info for RPi-Monitor

# Get processor model
echo "processor_model=$(cat /proc/cpuinfo | grep 'model name' | head -1 | sed 's/.*: //')"

# Get Linux distribution
if [ -f /etc/os-release ]; then
    . /etc/os-release
    echo "distribution=${PRETTY_NAME}"
else
    echo "distribution=Unknown"
fi

# Get kernel version
echo "kernel_version=$(uname -srm)"

# Get firmware version (RPi specific)
if [ -f /proc/device-tree/model ]; then
    echo "firmware=$(cat /proc/device-tree/model | tr -d '\0')"
else
    echo "firmware=Unknown"
fi

# Get package updates
echo "packages=0 upgradable(s)"
```

Stiskni CTRL+X, poté Y a Enter pro uložení souboru.

Nastavení práv pro spušt?ní skriptu

```
sudo chmod +x /opt/rpi-monitor/data/system_info.sh
```

3. Vytvoření docker-compose.yml

Do portaineru - stack, pojmenuj rpi-monitor a vloz kod:

```
version: '3.8'
services:
  rpi-monitor:
    image: michaelmiklis/rpi-monitor:latest
    container_name: rpi-monitor
    ports:
      - "8888:8888"
    volumes:
      - /boot:/boot:ro
      - /sys:/dockerhost/sys:ro
      - /proc:/dockerhost/proc:ro
      - /etc:/dockerhost/etc:ro
      - /usr/lib:/dockerhost/usr/lib:ro
      - /dev:/dev:ro
      - /var/run/docker.sock:/var/run/docker.sock:ro
      - /opt/rpi-monitor/data:/data
      - /opt/rpi-monitor/conf:/etc/rpimonitor/conf.d:ro
    environment:
      - TZ=Europe/Prague
    restart: unless-stopped
    privileged: true
    network_mode: bridge
```

5. Ov??ení funk?nosti

1. Otevři webový prohlížeč
2. Zadej adresu `http://IP_ADRESA_RPI:8888`
3. Měl by se ti zobrazit dashboard RPi-Monitor s plně funkčními daty