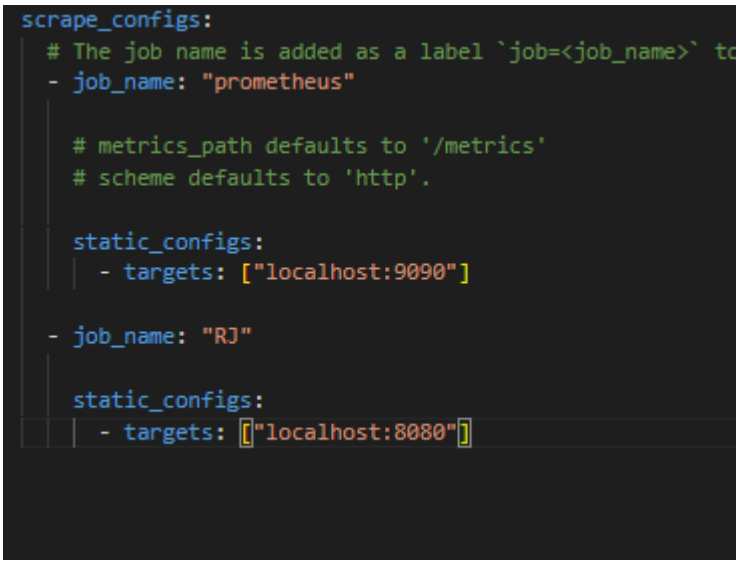
به نام خدا

امیررضا رجبی گزارش پروژه شبکه
9831126

تنظیمات فایل yaml :



قطعه کد agent :

agent.py

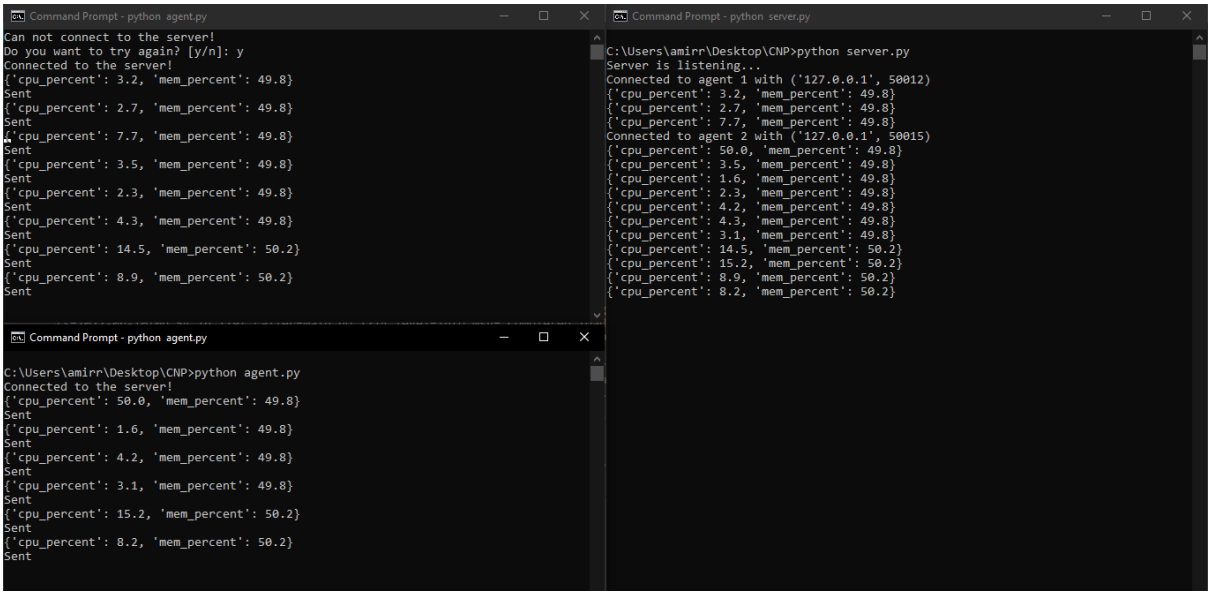
```python
# agent program to communicate with the server for get and send data to server by json and socket

# import libraries
import socket
import json
import time
import random
import psutil


class Agent:
    def __init__(self, HOST="127.0.0.1", PORT=8081):
        self.HOST = HOST
        self.PORT = PORT
        self.S = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    def connect(self):
        self.S.connect((self.HOST, self.PORT))

    def get_sys_data(self):
        data = {
            "cpu_percent": psutil.cpu_percent(),
            "mem_percent": psutil.virtual_memory().percent
        }
        print(data)
        return data

    def send(self):
        self.S.send(json.dumps(self.get_sys_data()).encode('ascii'))
        print("Sent")


agent = Agent()

until_connected = False
while not until_connected:
    try:
        agent.connect()
        until_connected = True
        print("Connected to the server!")
    except ConnectionRefusedError:
        print("Can not connect to the server!")
        yes = input("Do you want to try again? [y/n]: ")
        if yes != "y":
            print("Agent program closed!")
            quit()

try:
    while True:
        try:
            agent.send()
        except :
            print("Can not send data to the server!")
        time.sleep(1)
except:
    print("Problem in sending data!")
```

قطعه کد server:

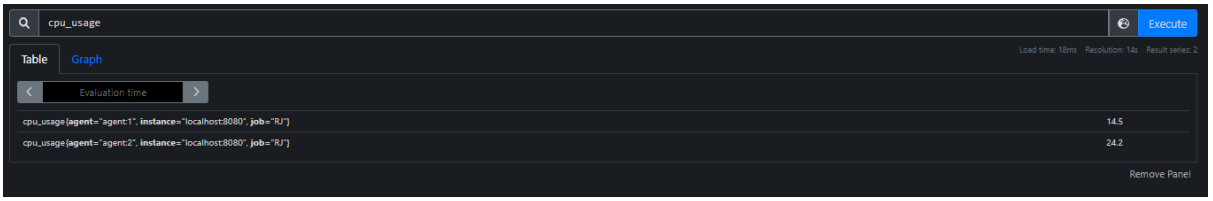
server.py

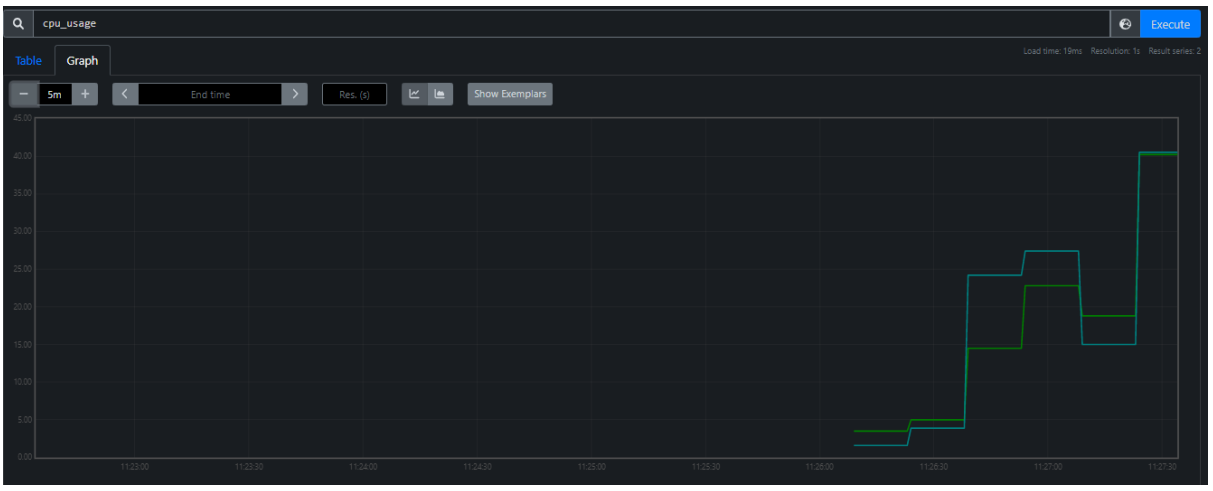
```python
"""
server program to communicate with the multiple agents for get and send data to agent by json and socket
server using prometheus
"""

# import libraries
import socket
from prometheus_client import start_http_server, Gauge
import _thread
import threading
import time
import json

class Server:
    def __init__(self, HOST="127.0.0.1", PORT=8081):
        start_http_server(8888)
        self.HOST = HOST
        self.PORT = PORT
        self.LOCK = threading.Lock()
        self.CPU = Gauge("cpu_usage", 'Usage of CPU', ["agent"])
        self.VMP = Gauge("virtual_memory_percent", 'virtual memory percent', ["agent"])
        self.agents = dict()
        self.num_agents = 0
        self.S = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.S.bind((self.HOST, self.PORT))

    def send(self, message, agent_id):
        data = json.loads(message)
        agent_name = f"agent:{agent_id}"
        print(data)
        self.CPU.labels(agent=agent_name).set(data['cpu_percent'])
        self.VMP.labels(agent=agent_name).set(data['mem_percent'])

    def up_server(self):
        self.S.listen(10)
        print("Server is listening...")
        while True:
            c, addr = self.S.accept()
            self.num_agents += 1
            self.agents[c] = self.num_agents
            self.LOCK.acquire()
            print(f"Connected to agent {self.num_agents} with {addr}")
            self.LOCK.release()
            _thread.start_new_thread(self.handle_agent, (c,))

    def handle_agent(self, c):
        agent_id = self.agents[c]
        try:
            while True:
                data = c.recv(1024)
                self.send(data, agent_id)
        except Exception as e:
            print(e)
            del self.agents[c]
            self.LOCK.acquire()
            print(f"Agent {agent_id} is disconnected from server")
            print(f"Total agents: {len(self.agents)}")
            self.LOCK.release()

server = Server()
server.up_server()
```

اجرای کد :



Metric cpu_usage:





Metric virtual_memory_percent: