Proiect retele

Wazuh: sistem de detectare intruziuni. Alertele sunt trimise catre botul creat de mine si permit echipei sa reactioneze rapid la incidentele de securitate.

Ansible si Semahore UI automatizeaza mentenanta. Playbookurile instaleaza software necesar pe toate masinile virtuale create asigurand uniformitatea sistemului.

Promotheus / Grafana: colecteaza metrici de perfomanta: dashboraduri in timp real, starea de sanatate etc

Uptime Kuma: monitorizeaza disponibilitatea serviciilor web. daca un serviciu ca gitlab pica, sistemul va notifica echipa prin botul creat.

Implementare

Ansible si Semaphore UI

Ansible: citeste playbookurile si executa automat acele comenzi pe toate masinile din retea

Playbookurile: actualizarea sistemelor Linux, Instalare vs code, dicker sau agenti de monitorizare

Semaphore UI: interfata web care permite apasarea unui buton si aplicarea acelor playbookuri in loc de comenzi din terminal

Wazuh

Wazuh Server & Indexer: primeste informatii de la toate calculatoarele si analizeaza daca se intampla ceva suspect

Wazuh Agent: un program pe care ansible il instaleaza pe statiile agentilor si raporteaza tot ce vede catre serverul central

Wazuh Bot: detecteaza un atac, trimite imediat mesaj la Mattermot pt avertizare

Grafana + Prometheus (sau Zabbix)

- se ocupa de performanta
- Prometheus / Zabbix: colecteaza date tehnice: cat de incarcat este procesorul, cata memorie RAM mai e libera
- Agent: “senzori” instalati pe calculatoare care trimit datele catre server
- Grafana: program care face dashboarduri din datele primite de mai sus

Uptime Kuma

Verifica non stop daca siteurile sau serviciile sunt online

trimite un semnal ping catre server, daca aceste nu raspunde il considera picat

implementare ansible si semaphore ui

- folosesc baza de date postgres pentru semaphore
- semaphore este interfata pentru ansible
- volume pentru persistenta de date
- folder ansible/ pentru playbookuri
- am facut un docker-compose unde am 2 containere:
  - semaphore pe portul 3000 (localhost:3000)
  - postgres (baza de date): am facut un volum pentru baza de date

    - am o retea virtuala izolata
- folder playbooks:
    - update_repos: face yum update → toate sistemele la zi
    - install_vscode: instaleaza VSCODE si adauga cheile de securitate si instaleaza pachetul code
    - install_docker: instaleaza Docker pe statii, face pull pe imaginile defualt cerute: hello-world, nginix, alpine
    - install_wazuh_agent: instaleaza Wazuh agent care trimite datele catre serverul Wazuh pe care il vom instala ulterior. Configureaza automat ca agentul sa stie IP-ul serverului (variabila MANAGER IP)
    - install_node: instaleaza node exporter pt Prometheus pt a expune metrici despre sistem: CPU, memorie, disc, pe un port de unde Promethreus le va colecta pentru a face graficele din Grafana

  COMENZI NECESARE:

  docker-compose up -d

  docker-compose -f docker-compose-test.yml up -d --build

  (cu ultima comanda pornesc 3 containere de Ubuntu ca sa simulez 2 statii si un server)

  comanda ca sa rulez un playbook:
  docker exec semaphore ansible-playbook -i /ansible/inventory/hosts.ini /ansible/playbooks/update_repos.yml

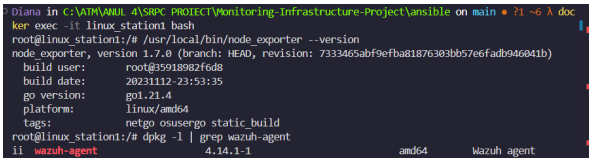
  output:

```
Diana in C:\ATM\ANUL 4\SRPC PROIECT\Monitoring-Infrastructure-Project on main ● ~4 λ docker exec semaphore ansible
-playbook -i /ansible/inventory/hosts.ini /ansible/playbooks/update_repos.yml

PLAY [Update repositories] *****************************************************

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host linux_station2 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
[WARNING]: Platform linux on host linux_station1 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [linux_station2]
ok: [linux_station1]

TASK [Update apt cache and upgrade packages (Debian/Ubuntu)] ******************
ok: [linux_station2]
ok: [linux_station1]

TASK [Update yum packages (RHEL/CentOS)] ***************************************
skipping: [linux_station1]
skipping: [linux_station2]

PLAY RECAP *********************************************************************
linux_station1             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

linux_station2             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

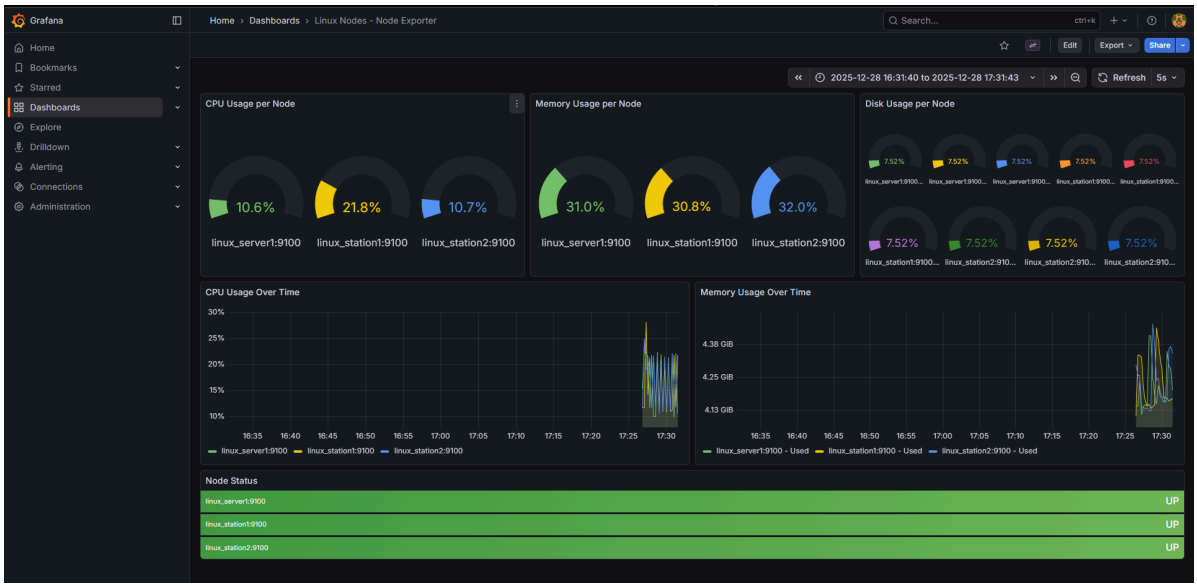
am facut un script care sa ruleze toate playbookurile, rulam cu comanda: `docker exec semaphore ./ansible/run_all.sh`

** problema e ca nu pot instala docker fiindca eu deja sunt intr un container de docker si e mai complicat, pe masina virtuala ar trebui sa mearga

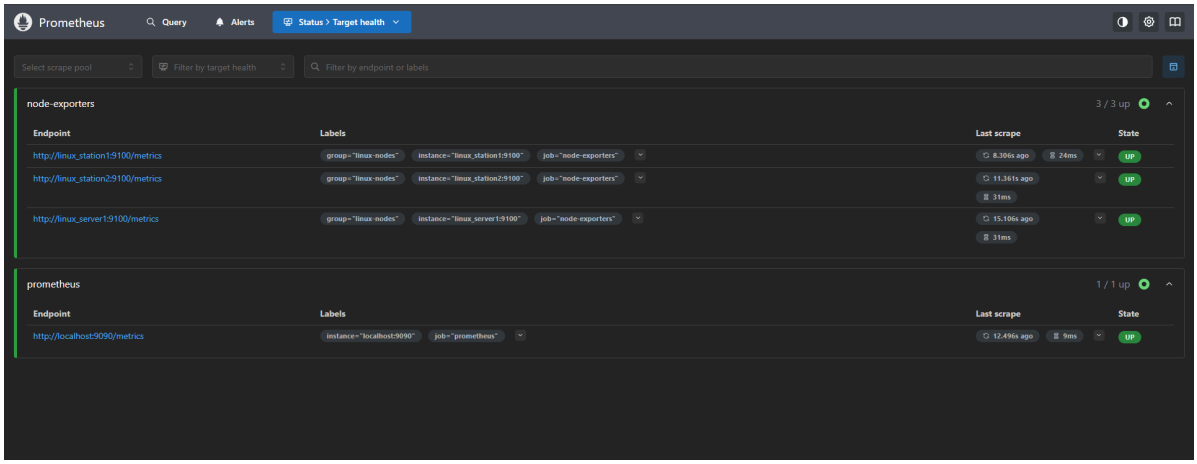
verificare instalare pe containere:



configurare Graphana

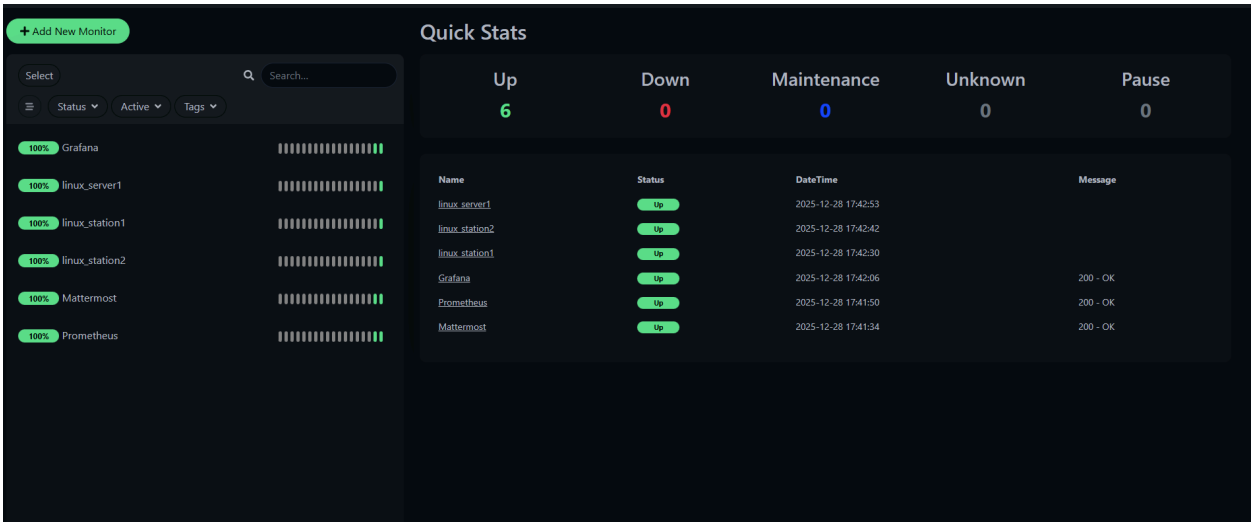


configurare Prometheus

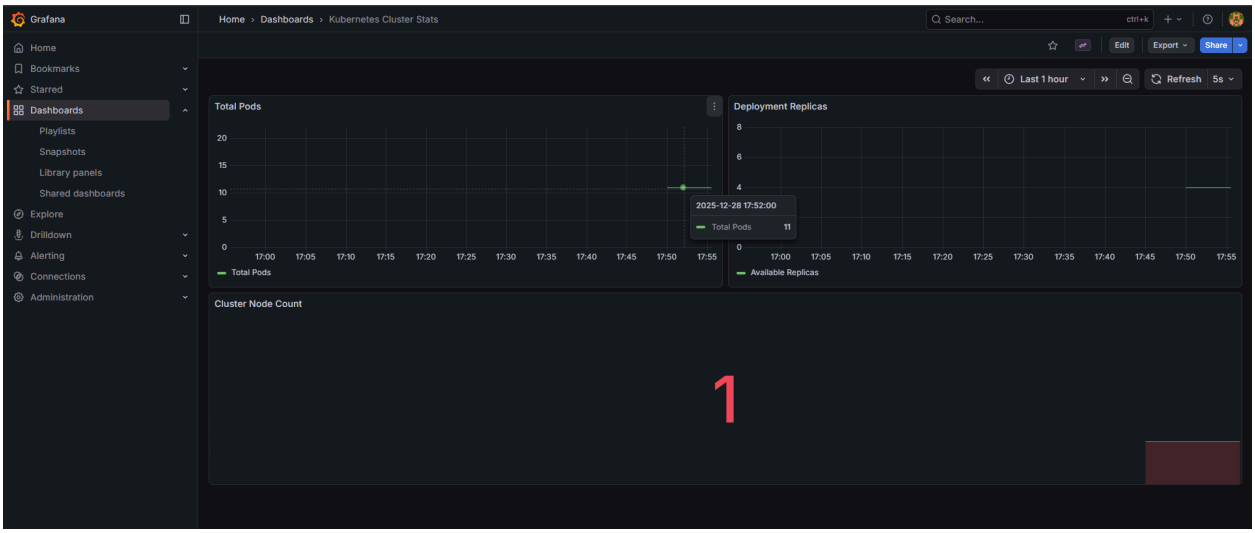


configurare Kuma

Monitor	Type	Target
Mattermost	HTTP	`http://mattermost:8065`
Prometheus	HTTP	`http://prometheus:9090`
Grafana	HTTP	`http://grafana:3000`
linux_station1	Ping	`linux_station1`
linux_station2	Ping	`linux_station2`
linux_server1	Ping	`linux_server1`



graphana kubernetes



probleme intampinate si alternative

1. OpenSearch

- am folosit opensearch in loc de wazuh indexer pentru compatibilitate (cel default n a mers) ;
- am expus portul 9200 ;
- cu opensearch stochez si indexez toate alertele si log-urile primite de la Wazuh Manager

```
wazuh.indexer:
  image: opensearchproject/opensearch:2.11.0
  ports: "9200:9200"
  environment:
    - DISABLE_SECURITY_PLUGIN=true
```

2. Wazuh Manager

```
wazuh.manager:
  image: wazuh/wazuh-manager:4.7.0
  ports:
    - "1514:1514"  # Agent communication
    - "1515:1515"  # Agent enrollment
    - "55000:55000" # API
```

- este practic nucleul wazuh
- primeste datele de la agentii de pe fiecare container
- 1514 pt trimitere evenimente
- 1515 pt inregistrarea agentilor

3. wazuh dashboard

```
wazuh.dashboard:
  image: wazuh/wazuh-dashboard:4.7.0
  ports: "443:5601"
```

- interfata web unde vad dashboardul si vizualizarea altertelor
- accesibil pe https://localhost:443

4. integrare Mattermost Bot
    a. am folosit postgreSQL pentru Mattermost ca baza de date
    b. este configurat in ossec.config si functioneaza prin hookuri:
        i. managerul detecteaza un eveniment
        ii. trimite post request la webhookul mattermost
        iii. mattermost afiseaza alerta
    c. accesibil la http://localhost:8065
    d. NU MI A PORNIT wazuh-db DECI N AM PUTUT TESTA
5. Ansible Playbooks
    a. aici am instalat code-server, nu vs code cum era cerinta pt ca era ceva mai complicat pe docker sa pun vs code fara interfata, asa ca am pus

serverul si a functionat

b. cu celelalte Nu am avut probleme

6. Semaphore UI
   a. stocheaza rezultatele in loguri in postgresql
   b. accesibil la http://localhost:3000
   c. admin admin
7. promtheus + grafana
   a. accesibil la http://localhost:9090
   b. joburi configurate: promtheus, node-exportes, kube-state-metrics
   c. grafana e accesibila la http://localhost:3001 cu admin admin
   d. in grafana am configurat 2 dashboarduri:
      i. node exporter dashboard unde am CPU USAGE, MEMORY USAGE ETC
      ii. kubernetes dashboard unde am metrici pt pod count per namesapce, storage usage

```
# CPU Usage
100 - (avg(irate(node_cpu_seconds_total{mode="idle"}[5m])) * 100)

# Memory Usage
(node_memory_MemTotal_bytes - node_memory_MemAvailable_bytes) /
node_memory_MemTotal_bytes * 100

# Disk I/O
rate(node_disk_read_bytes_total[5m])
```

8. Uptime Kuma
   a. accesibil la http://localhost:3002
   b. am atasat poza mai sus cu ce am configurat acolo (poza cu tabelul)
9. Cluster Kubernetes
   a. am folosit kind pt cluster local
   b. am deployat kube-state-metrics pt metrici K8s
   c. portul 30000 expus pt prometheus
   d. Dashboard în Grafana cu: Total Pods, Deployment Replicas, Node Count