

# 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 Semaphore UI automatizeaza mentenanta. Playbookurile instaleaza software necesar pe toate masinile virtuale create asigurand uniformitatea sistemului.

Promotheus / Grafana: colecteaza metrii de performanta: 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 analogii

### Ansible si Semaphore UI

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

Playbookurile: actualizarea sistemelor Linux, Instalare vs code, docker 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: primește informații de la toate calculatoarele și analizează dacă se întâmplă ceva suspect

Wazuh Agent: un program pe care Ansible îl instalează pe stațiile agentilor și raportează tot ce vede către serverul central

Wazuh Bot: detectează un atac, trimită imediat mesaj la Mattermost pt avertizare

## Grafana + Prometheus (sau Zabbix)

- se ocupă de performanță
- Prometheus / Zabbix: colectează date tehnice: cât de încarcat este procesorul, câtă memorie RAM mai e liberă
- Agent: "senzori" instalati pe calculatoare care trimit datele către server
- Grafana: program care face dashboarduri din datele primite de mai sus

## Uptime Kuma

Verifică non stop dacă siteurile sau serviciile sunt online

trimite un semnal ping către server, dacă acestea nu răspund îl consideră picat

## Implementare Ansible și Semaphore UI

- folosesc bază de date PostgreSQL pentru Semaphore
- Semaphore este interfața pentru Ansible
- volume pentru persistență de date
- folder Ansible/ pentru playbookuri
- am făcut un Docker-Compose unde am 2 containere:
  - Semaphore pe portul 3000 (localhost:3000)
  - PostgreSQL (bază de date): am făcut 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 VS CODE si adauga cheile de securitate si instaleaza pachetul code
  - install\_docker: instaleaza Docker pe statii, face pull pe imaginile defualt cerute: hello-world, nginx, alpine
  - install\_wazuh\_agent: instaleaza Wazuh agent care trimit 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 metrii despre sistem: CPU, memorie, disc, pe un port de unde Prometheus 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 PROJECT\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 container:

```
Diana in C:\ATM\ANUL 4\SRPC PROJECT\Monitoring-Infrastructure-Project\ansible on main • ?1 ~6 λ doc
ker exec -it linux_station1 bash
root@linux_station1:/# /usr/local/bin/node exporter --version
node exporter, version 1.7.0 (branch: HEAD, revision: 7333465abf9efba81876303bb57e6fadb946041b)
  build user: root@35918982fd8
  build date: 2023-11-23:53:35
  go version: go1.21.4
  platform: linux/amd64
  tags: netgo osusergo static_build
root@linux_station1:/# dpkg -l | grep wazuh-agent
ii  wazuh-agent                         4.14.1-1                                amd64        wazuh agent
```

