

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.

Prometheus / Grafana: colecteaza metrice de performanta: dashboards 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: citește 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 agenților și raportează tot ce vede către serverul central

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

Grafana + Prometheus (sau Zabbix)

- se ocupă de performanță
- Prometheus / Zabbix: colectează date tehnice: cât de încărcat este procesorul, câtă memorie RAM mai e liberă
- Agent: "senzori" instalați 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 baza de date postgres pentru semaphore
- semaphore este interfata pentru ansible
- volume pentru persistența de date
- folder ansible/ pentru playbookuri
- am făcut 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 default cerute: hello-world, nginx, 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 metrice 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:

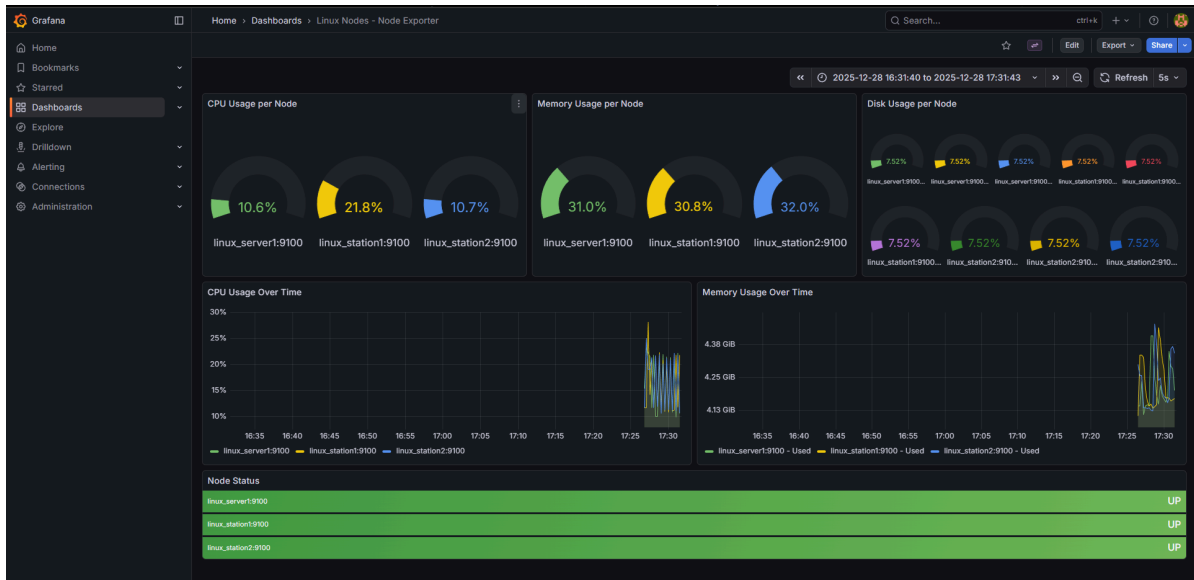
docker exec semaphore ./ansible/run_all.sh

** 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 containere:

```
Diana in C:\ATM\ANUL 4\SRPC PROJECT\Monitoring-Infrastructure-Project\ansible on main • 71 ~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: 7333465abf9efba81876383b57cefadb946041b)
  build user:   root@35918982f6da
  build date:   20231112-23:53:35
  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
```

configurare Graphana



configurare Prometheus

The image shows the Prometheus Status page, which displays the health of the targets being scraped. The page is divided into two sections: "node-exporters" and "prometheus".

Section	Endpoint	Labels	Last scrape	State
node-exporters	http://linux_station1:9100/metrics	group="linux-nodes", instance="linux_station1:9100", job="node-exporters"	8.306s ago, 24ms	UP
	http://linux_station2:9100/metrics	group="linux-nodes", instance="linux_station2:9100", job="node-exporters"	11.361s ago, 31ms	UP
	http://linux_server1:9100/metrics	group="linux-nodes", instance="linux_server1:9100", job="node-exporters"	15.106s ago, 31ms	UP
prometheus	http://localhost:9090/metrics	instance="localhost:9090", job="prometheus"	12.496s ago, 9ms	UP

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

+ Add New Monitor

Select

Search...

≡

Status

Active

Tags

100% Grafana

100% linux_server1

100% linux_station1

100% linux_station2

100% Mattermost

100% Prometheus

Quick Stats

Up

Down

Maintenance

Unknown

Pause

6

0

0

0

0

Name	Status	DateTime	Message
linux_server1	Up	2025-12-28 17:42:53	
linux_station2	Up	2025-12-28 17:42:42	
linux_station1	Up	2025-12-28 17:42:30	
Grafana	Up	2025-12-28 17:42:06	200 - OK
Prometheus	Up	2025-12-28 17:41:50	200 - OK
Mattermost	Up	2025-12-28 17:41:34	200 - OK

graphana kubernetes

