Supervision de Guacamole avec Nagios dans un Conteneur Docker sur Ubuntu

On va mettre en place un serveur de supervision Nagios dans un conteneur Docker sur Ubuntu pour surveiller Guacamole.

Plan d'action

1. Introduction à Nagios et Docker
2. Installation de Docker et Docker Compose sur Ubuntu
3. Déploiement de Nagios dans un conteneur
4. Configuration de Nagios pour superviser Guacamole
5. Ajout de services et vérifications

Introduction à Nagios et Docker

Nagios est une solution open-source de supervision permettant de surveiller les serveurs, services et équipements réseau. Il peut alerter en cas de panne et fournir des métriques sur l'état des machines.

Docker permet de déployer des applications dans des conteneurs légers, facilitant l'installation et la maintenance des services comme Nagios.

L'objectif ici est de superviser Guacamole en utilisant Nagios exécuté dans un conteneur Docker.

Installation de Docker et Docker Compose sur Ubuntu

```
zafar@auth-srv:~$ sudo apt install -y docker.io
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
  libllvm17t64
```

```
zafar@auth-srv:~$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
zafar@auth-srv:~$
```

On a activé et démarré Docker puis on ajoute un utilisateur au group Docker qui exécute commande suo san mot de passe puis on installe docker compose et enfin on vérifie la version de composer

```
zafar@auth-srv:~$ sudo systemctl enable --now docker
zafar@auth-srv:~$ sudo usermod -aG docker $USER
zafar@auth-srv:~$ newgrp docker
zafar@auth-srv:~$ sudo apt install -y docker-compose
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
```

```
zafar@auth-srv:~$ docker-compose --version
docker-compose version 1.29.2, build unknown
zafar@auth-srv:~$
```

Déploiement de Nagios dans un Conteneur Docker

On crée un répertoire pour Nagios et un fichier `docker-compose.yml` pour Nagios

```
zafar@auth-srv:~$ mkdir -p ~/nagios && cd ~/nagios
zafar@auth-srv:~/nagios$ nano docker-compose.yml
zafar@auth-srv:~/nagios$
```

```
  GNU nano 7.2                                  docker-compose.yml *
version: '3'
services:
  nagios:
    image: jasonrivers/nagios
    container_name: nagios
    restart: always
    ports:
      - "8080:80"
    volumes:
      - ./nagios/etc:/opt/nagios/etc
      - ./nagios/var:/opt/nagios/var
      - ./nagios/logs:/var/log/nagios
      - ./nagios/plugins:/opt/Custom-Nagios-Plugins
    environment:
      - NAGIOSADMIN_USER=admin
      - NAGIOSADMIN_PASS=admin
```

On lance e conteneur

```
zafar@auth-srv:~/nagios$ docker-compose up -d
Creating network "nagios_default" with the default driver
Pulling nagios (jasonrivers/nagios:)...
latest: Pulling from jasonrivers/nagios
ff65ddf9395b: Pulling fs layer
785b9873bdf4: Pulling fs layer
785b9873bdf4: Extracting [================================>                  ]  181.6MB/275.3MB
53aff88babc4: Download complete
d72f92e29533: Download complete
706ed7d4ce0a: Download complete
```

```
9a90645e352c: Pull complete
8e911c59da28: Pull complete
c219d58cc3f9: Pull complete
b0e280e9aa8c: Pull complete
8c389e58e867: Pull complete
Digest: sha256:2a7c2b20d118baf92b47b69a3901e68dd7664617801b94e560bc4d6564d6ae54
Status: Downloaded newer image for jasonrivers/nagios:latest
Creating nagios ... done
zafar@auth-srv:~/nagios$
```

Nagios fonctionne

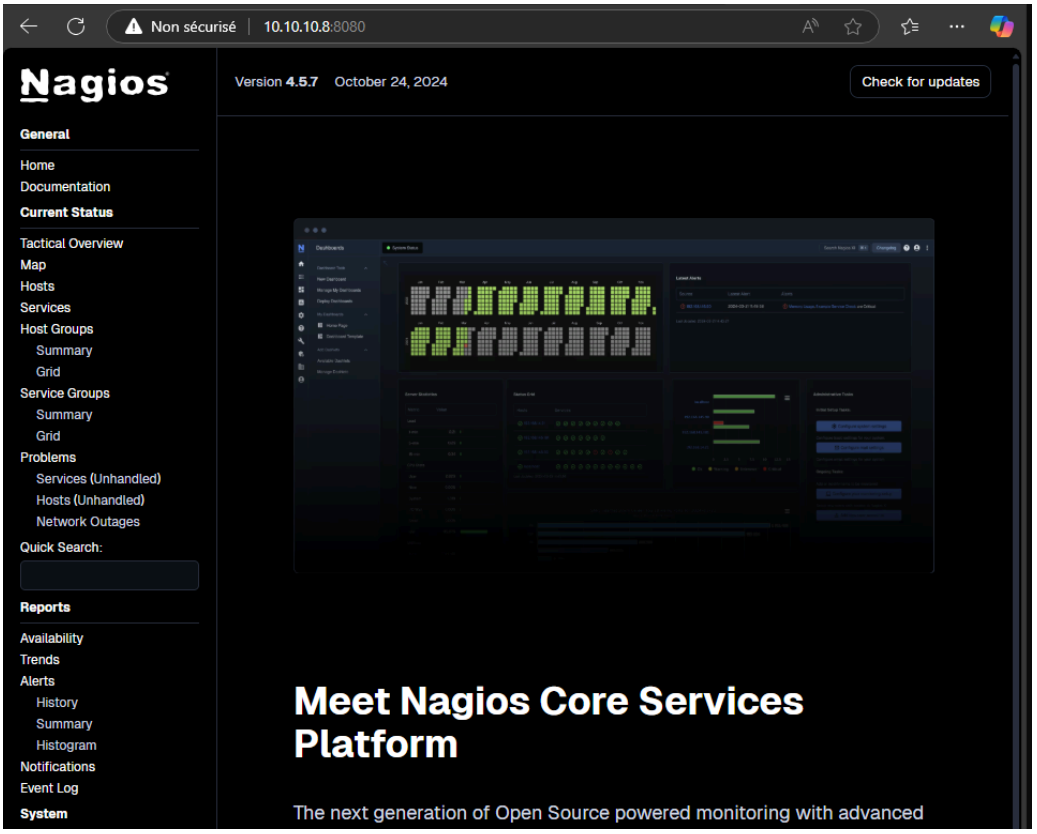
```
zafar@auth-srv:~/nagios$ docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED              STATUS              PORTS                                                NAMES
b7dd8225b84e   jasonrivers/nagios   "/usr/local/bin/star…"   About a minute ago   Up About a minute   5667/tcp, 0.0.0.0:8080->80/tcp, :::8080->80/tcp   nagios
zafar@auth-srv:~/nagios$
```

Accéder à l'interface Web de Nagios

👉 **http://=10.10.10.8:8080**

👤 **Utilisateur :** `admin`

🔑 **Mot de passe :** `admin`



Configuration de Nagios pour Superviser Guacamole

Nous allons maintenant configurer **Nagios** pour superviser **Guacamole**. On va utiliser le protocole **HTTP** pour vérifier si l'interface Web de Guacamole est accessible.

On accède au conteneur Nagios

```
zafar@auth-srv:~/nagios$ docker exec -it nagios /bin/bash
root@b7dd8225b84e:/#
```

On édite le fichier de configuration des hôtes

```
zafar@auth-srv:~/nagios$ docker exec -it nagios /bin/bash
root@b7dd8225b84e:/# nano /opt/nagios/etc/objects/guacamole.cfg
bash: nano: command not found
root@b7dd8225b84e:/# apt update && apt install nano -y
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
```

```
  GNU nano 7.2                    /home/zafar/nagios/nagios/etc/objects/guacamole.cfg *
define host {
    use linux-server
    host_name guacamole-server
    alias Guacamole Server
    address 10.10.10.4  ; Mets l'IP correcte de ton serveur Guacamole
    max_check_attempts 5
    check_period 24x7
    notification_interval 30
    notification_period 24x7
}

define service {
    use generic-service
    host_name guacamole-server
    service_description HTTP Web Interface
    check_command check_http
}
```

On Inclure ce fichier dans la config Nagios

```
root@b7dd8225b84e:/# echo 'cfg_file=/opt/nagios/etc/objects/guacamole.cfg' >> /opt/nagios/etc/nagios.cfg
root@b7dd8225b84e:/#
```

Puis on redémarre le service

```
zafar@auth-srv:~/nagios$ docker restart nagios
nagios
zafar@auth-srv:~/nagios$ docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED          STATUS         PORTS
                NAMES
b7dd8225b84e   jasonrivers/nagios   "/usr/local/bin/star…"   10 minutes ago   Up 5 seconds   5667/tcp, 0.0.0.0:8080->80/tcp,
:::8080->80/tcp   nagios
```

Fichier des commande

```
root@b7dd8225b84e:/# nano /opt/nagios/etc/objects/commands.cfg
root@b7dd8225b84e:/#
```

```
zafar@auth-srv:~/nagios$ docker exec -it nagios /bin/bash
root@b7dd8225b84e:/# apt update && apt install -y nagios-plugins
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
```

```
root@b7dd8225b84e:/# ls -l /usr/lib/nagios/plugins/check_http
-rwxr-xr-x 1 root root 81608 Apr  1  2024 /usr/lib/nagios/plugins/check_http
root@b7dd8225b84e:/# /usr/lib/nagios/plugins/check_http -H 10.10.10.4
HTTP OK: HTTP/1.1 200 - 3127 bytes in 0.002 second response time |time=0.002388s;;;0.000000;10.000000 size=3127B;;;0;
```

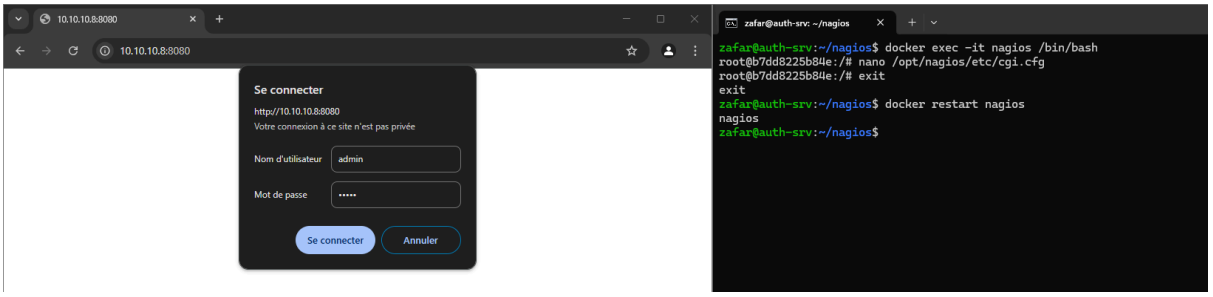
Droit :

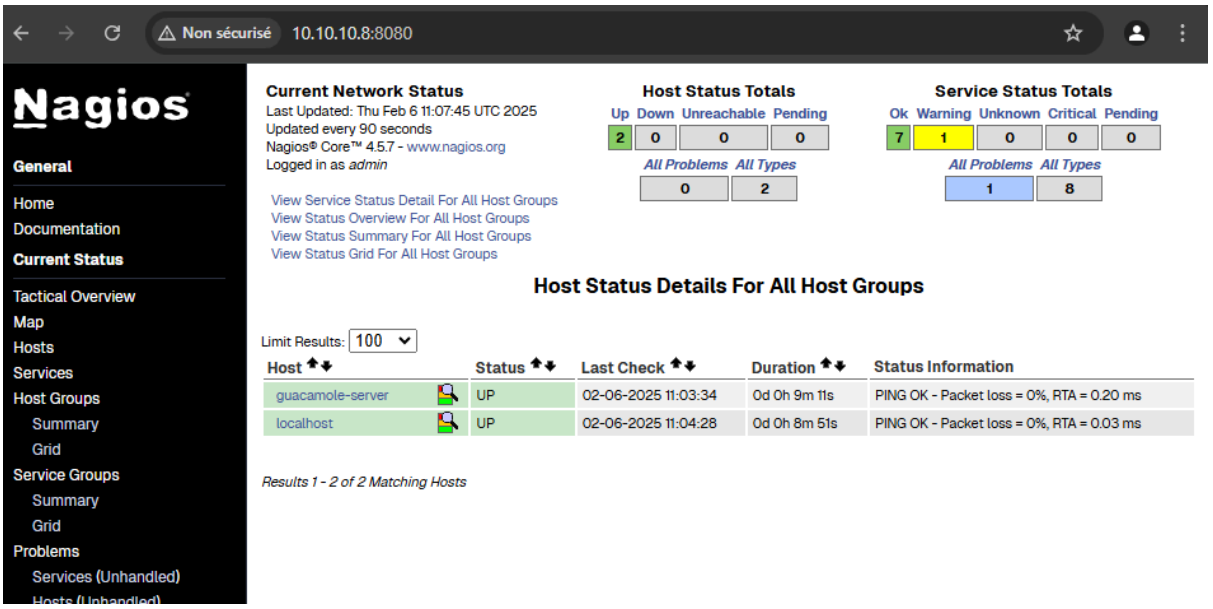
```
zafar@auth-srv:~/nagios$ docker exec -it nagios /bin/bash
root@b7dd8225b84e:/# chown -R nagios:nagios /opt/nagios/var
root@b7dd8225b84e:/# chmod -R 775 /opt/nagios/var
root@b7dd8225b84e:/#
```

```
root@b7dd8225b84e:/# exit
exit
zafar@auth-srv:~/nagios$ docker restart nagios
nagios
zafar@auth-srv:~/nagios$
```

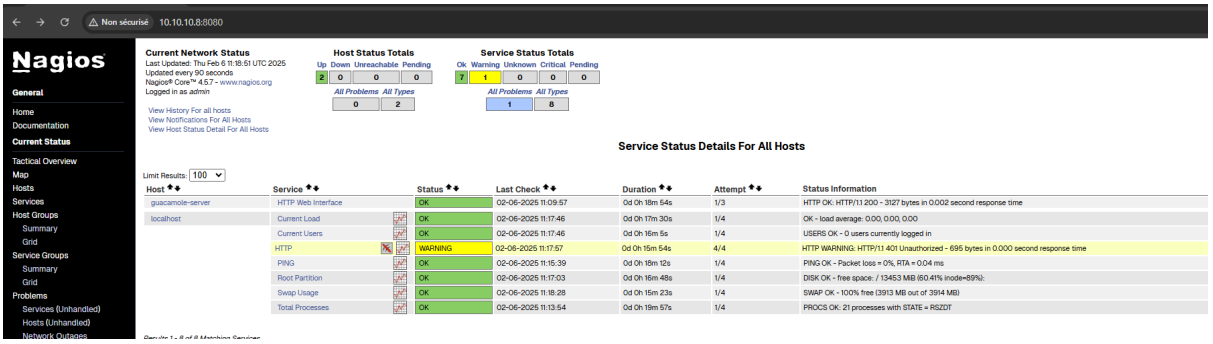
Authorisation de tous les service et configuration pour user admin

```
 GNU nano 7.2                                  /opt/nagios/etc/cgi.cfg


authorized_for_all_services=admin
authorized_for_all_hosts=admin



# GLOBAL HOST/SERVICE COMMAND ACCESS
# These two options are comma-delimited lists of all usernames that
# can issue host or service related commands via the command
# CGI (cmd.cgi) for all hosts and services that are being monitored.
# By default, users can only issue commands for hosts or services
# that they are contacts for (unless you you choose to not use
# authorization).  You may use an asterisk (*) to authorize any
# user who has authenticated to the web server.

authorized_for_all_service_commands=admin
authorized_for_all_host_commands=admin
```





Super ! 🎉 mon serveur Guacamole est bien surveillé avec Nagios. Maintenant, on peut aller plus loin avec des configurations avancées.



Super ! 🎉 mon serveur Guacamole est bien surveillé avec Nagios. Maintenant, on peut aller plus loin avec des configurations avancées.

2️⃣ Vérifier si le processus Guacamole tourne

On cree notre commande

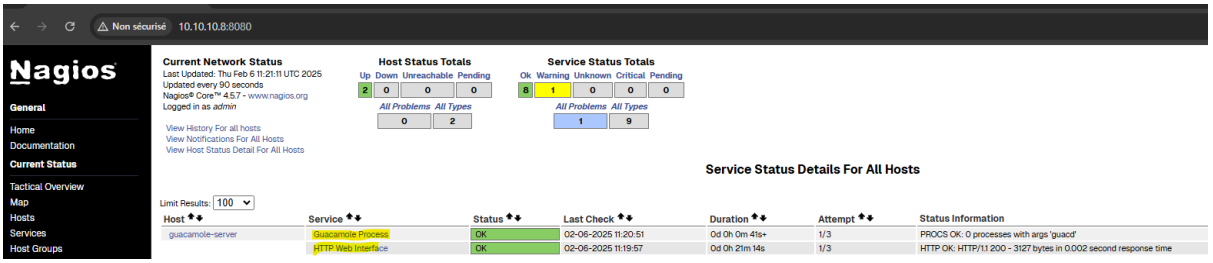
```
/opt/nagios/etc/objects/commands.cfg *
```

```
define command {
    command_name    check_procs
    command_line    $USER1$/check_procs -c $ARG1$ -a $ARG2$
}
```

```
/opt/nagios/etc/objects/guacamole.cfg *
```

```
define service {
    use                 generic-service
    host_name           guacamole-server
    service_description Guacamole Process
    check_command       check_procs!1!guacd
}
```

```
root@b7dd8225b84e:/# ls /usr/lib/nagios/plugins/check_procs
/usr/lib/nagios/plugins/check_procs
root@b7dd8225b84e:/# nano /opt/nagios/etc/objects/commands.cfg
root@b7dd8225b84e:/# nano /opt/nagios/etc/objects/guacamole.cfg
root@b7dd8225b84e:/# /opt/nagios/bin/nagios -v /opt/nagios/etc/nagios.cfg
```

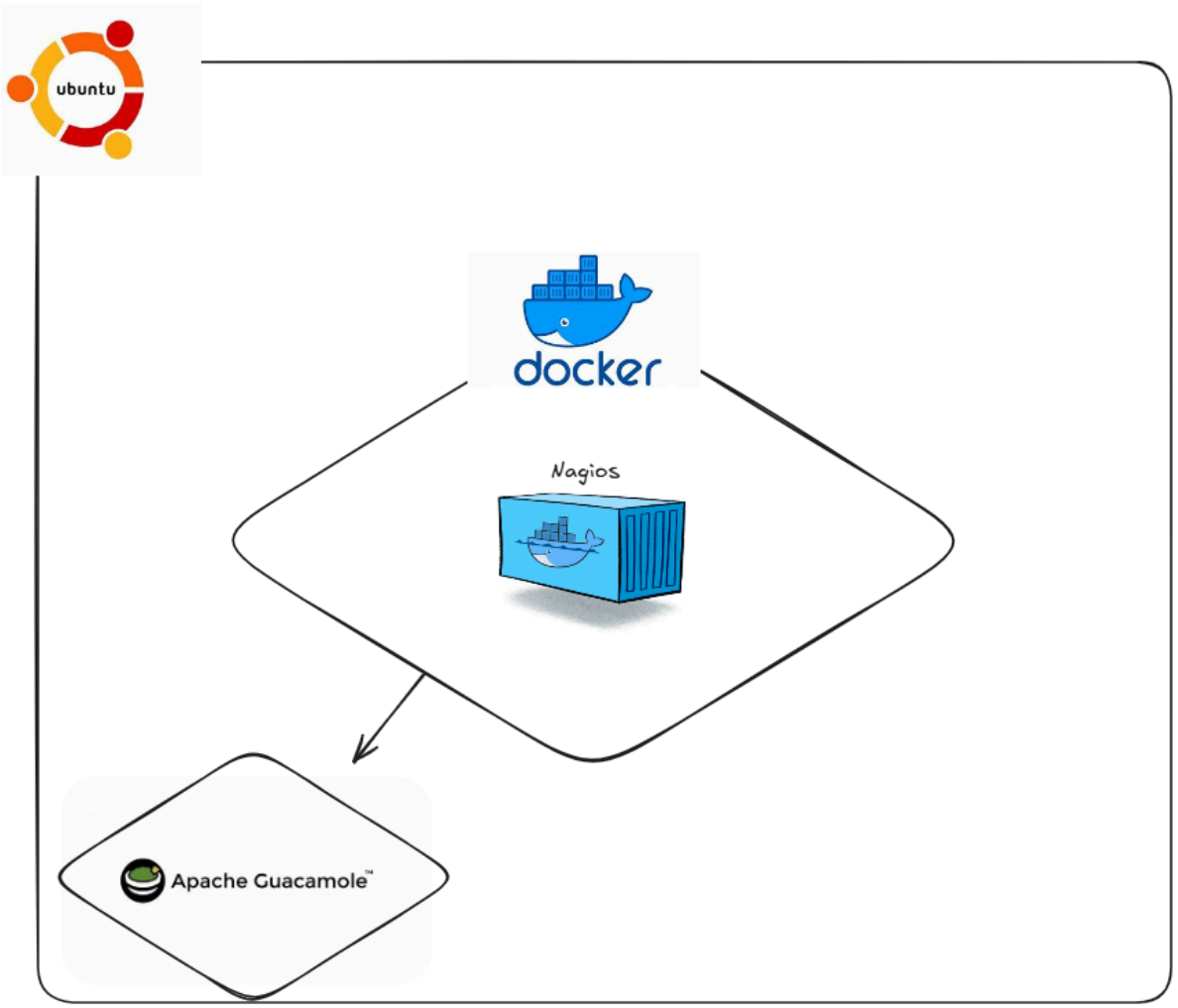


Super ! 🎉

```
define contact {

    contact_name            nagiosadmin             ; Short name of user
    use                     generic-contact         ; Inherit default values from generic-contact te
    alias                   Nagios Admin            ; Full name of user
    email                   suprot@daudruy.fr ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
}
```

Tous les service de supervision :

```
root@b7dd8225b84e:/# ls /opt/nagios/libexec/
check-mqtt.py   check_ftp           check_mailq             check_oracle        check_ssmtp
check_apt       check_game          check_mem.pl            check_overcr        check_swap
check_breeze    check_hpjd          check_mrtg              check_pgsql         check_tcp
check_by_ssh    check_http          check_mrtgtraf          check_ping          check_time
check_clamd     check_icmp          check_mssql_database.py check_pop           check_udp
check_cluster   check_ide_smart     check_mssql_server.py   check_procs         check_ups
check_dbi       check_ifoperstatus  check_nagios            check_real          check_uptime
check_dhcp      check_ifstatus      check_ncpa.py           check_rpc           check_users
check_dig       check_imap          check_nntp              check_sensors       check_vpn
check_disk      check_ircd          check_nntps             check_simap         check_wave
check_disk_smb  check_jabber        check_nrpe              check_smtp          mibs
check_dns       check_jenkins       check_nt                check_snmp          negate
check_dummy     check_ldap          check_ntp               check_spop          remove_perfdata
check_file_age  check_ldaps         check_ntp_peer          check_sql           urlize
check_flexlm    check_load          check_ntp_time          check_ssh           utils.pm
check_fping     check_log           check_nwstat            check_ssl_validity  utils.sh
root@b7dd8225b84e:/#
```


ubuntu
docker
Nagios
Apache Guacamole™