**Cacti** est système de supervision open-source largement utilisé pour surveiller les performances des réseaux informatiques.

**Cacti**

Fonctionnalités principales :

Cacti permet de collecter, stocker et présenter des données de performance réseau sous forme de graphiques, Il utilise le protocole SNMP (**Simple Network Management Protocol**) pour interroger les périphériques réseau et recueillir des informations telles que le trafic, l’utilisation du CPU, la bande passante, etc.

**Architecture** : Cacti repose sur une architecture client-serveur. Le serveur Cacti héberge l’application web (**Apache, Nginx, IIS**) et la base de données (**Maria DB, MySQL**) où sont stockées les données de performance. Les périphériques réseaux surveillés sont des clients SNMP qui envoient leurs données au serveur Cacti.

**Interface utilisateur** : Cacti propose une interface web conviviale permettant aux utilisateurs de visualiser les données de performances sous personnaliser les graphiques, créer des tableaux de bord et configurer des alertes en fonction des seuils (**thold**) de performance.

**Gestion des graphiques** : Cacti offre des outils pour créer et personnaliser des graphiques de performance. Les utilisateurs peuvent choisir parmi une variété de modèles de graphiques prédéfinis ou créer leurs propres modèles pour répondre à leurs besoins spécifiques.

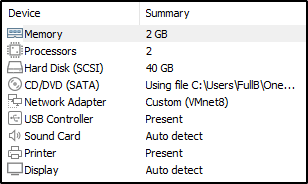
**Gestion de données** : Les données de performance collectées par Cacti propose des fonctionnalités de gestion des données telles que la sauvegarde, la restauration et l’archivage des données.

**Extensibilité** : Cacti est hautement extensible grâce à son architecture modulaire. Il prend en charge l’ajout de plugins et d’extensions pour étendre ses fonctionnalités de surveillance et d’analyse.

**Tableau comparatif des fonctionnalités.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fonctionnalité | Cacti | Zabbix | Wazuh | Prometheus |
| Type | Superviseur Réseau | Superviseur Réseau | Superviseur de sécurité et de conformité | Superviseur systèmes et services |
| Collecte de données | SNMP | SNMP, IPMI, JMX | Logs, événements de sécurité | Exporteurs, API, Services Intégrations |
| Interface utilisateur | Web | Web | Web | Web |
| Graphiques | Oui | Oui | Non | Oui |
| Personnalisation | Modèles de graphique personnalisables | Personnalisable | Personnalisable | Personnalisable |
| Alertes | Oui | Oui | Oui | Oui |
| Extensibilité | Plugins | Plugins, scripts, API | Règles de détection personnalisées | Exportateur intégrations tierces |
| Licence | Open-source | Open-source | Open-source | Open-source |

Installation de Cacti par ccontenteur sur un serveur Ubuntu 22.04 (Jammy jellfish) LTS

Caractéristique du serveur Ubuntu (pré-prod) :

Options VMWare WorkStation

* RAM 2 Go
* Processeurs 2
* Disque dur 40 Go
* Adaptateur réseau VMnet8 (192.168.8.0/24)

\*Ces informations sont modifiables selon vos besoin et envie.

Les commandes sur le serveur Ubuntu :

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquementAdresse statique sur le serveur Ubuntu

Netplan adressage statique

Mise a jour des repositories du serveur

* apt update
* apt ugrade

Installation de docker (docker doc : <https://docs.docker.com/engine/install/ubuntu/>)

Installer en utilisant le dépôt apt

# Ajouter la clé GPG officielle de Docker

* apt update
* apt-get install ca-certificates curl
* install -m 0755 -d /etc/apt/keyrings
* curl -fsSL [https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc](https://download.docker.com/linux/ubuntu/gpg%20-o%20/etc/apt/keyrings/docker.asc)
* chmod a+r /etc/apt/keyrings/docker.asc

# Ajouter le dépôt aux sources d'apt :

* echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

Installer les paquets Docker :

* sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

Vérifiez que l'installation de Docker Engine est réussie en exécutant l'image hello-world

* sudo docker run hello-world

Puis installer du container Cacti en par admindean (<https://hub.docker.com/r/admindean/cacti>)

Environnement d'exécution：

1 Pull image

* docker pull admindean/cacti :latest

2 Créer un répertoire de sauvegarde

* mkdir -p /usr/share/cacti/rra && chmod 777 -R /usr/share/cacti/rra
* mkdir -p /usr/share/cacti/weathermap && chmod 777 -R /usr/share/cacti/weathermap
* mkdir -p /usr/share/cacti/mysql\_backup && chmod 777 -R /usr/share/cacti/mysql\_backup

3. Exécuter l'instance de docker

* docker run -d --restart=always --name cacti --hostname cacti -p 2080:80 -p 20443:443 --privileged=true -v /usr/share/cacti/rra:/usr/share/cacti/rra -v /usr/share/cacti/weathermap:/usr/share/cacti/weathermap -v /usr/share/cacti/mysql\_backup:/usr/share/cacti/mysql\_backup admindean/cacti:latest

(Port en rouge : Numéros de port personnalisable)

(Port en bleu : Numéros de port du container à ne pas modifiée)

4. Mise à jour du fuseau horaire

* docker exec -it cacti /bin/bash
* cp /usr/share/zoneinfo/Europe/Paris/etc/localtime

5. Accès au web

* <http://your_ip/cacti> dans notre cas http://192.168.8.150:2080/cacti
* <https://your_ip/cacti> dans notre cas https://192.168.8.150:20443/cacti

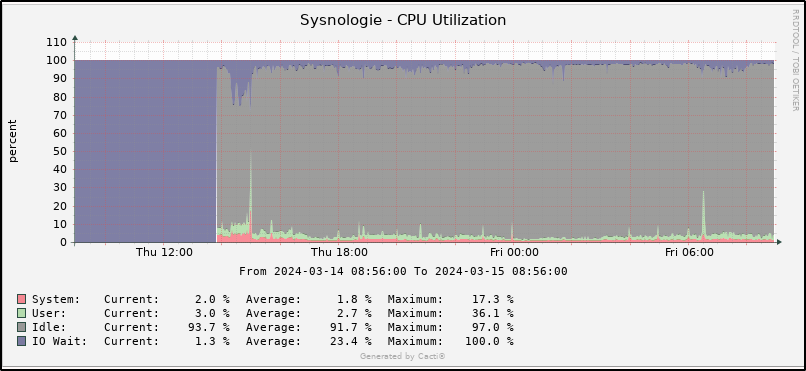
Première connexion :

* username: admin
* password: admin

6. Mot de passe MariaDB

* root / root
* cactiuser / cactiuser

L’interface graphique de cacti :



pfsnse CPU Utilisation

Graphique d'utilisation du processeur (CPU Utilization) : Comme décrit précédemment, ce graphique représente l'utilisation du processeur d'un système informatique sur une période de temps donnée, indiquant le pourcentage d'utilisation du processeur par rapport à sa capacité maximale.