



Analyse de l'implémentation d'un Pipeline CI/CD et d'un SIEM

Optimisation des Flux de Développement et Renforcement de la Sécurité



Sommaire

1. Pipeline CI/CD

3

2. SIEM



PIPELINE CI/CD

Continuous Integration et Continuous Déploiement

Objectifs CI/CD

Mettre en place sur 2 applis de la HAS

Solution de CI/CD :

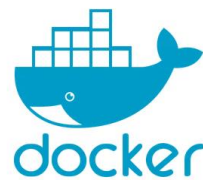
Conteneurisation des 2 applications: PACTE et SIAM2

Intégration continue (validation et construction) : nécessite que les développeurs aient des versions **régulières**.

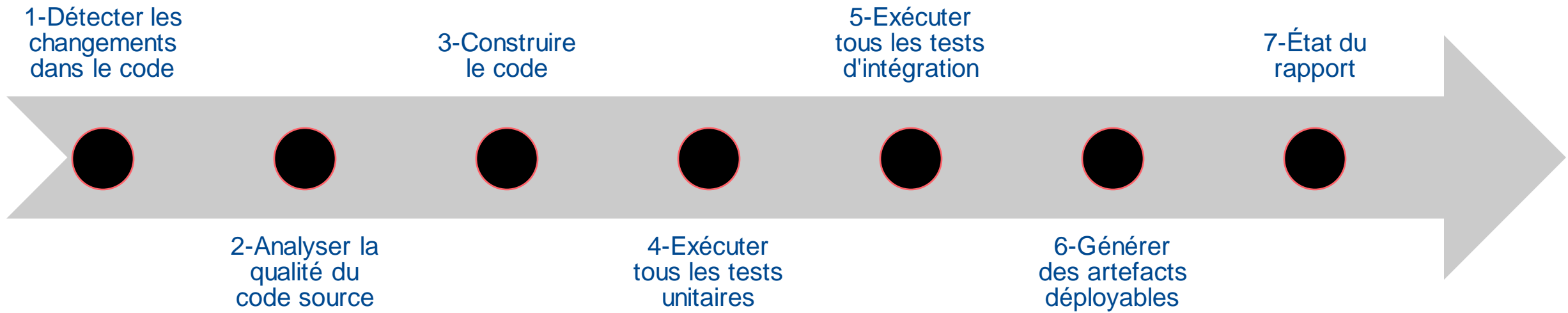
Livraison continue : **automatisation** des processus et les tests jusqu'à la **phase de déploiement** en environnement de **développement**, de préprod et de prod.

Déploiement continue : même phases que la **livraison continue** à la différence qu'elle permet le déploiement automatisé jusqu'à l'environnement de production **sans validation humaine**.

Avec GitLab, Docker, Jenkins



Etapes d'un pipeline CI



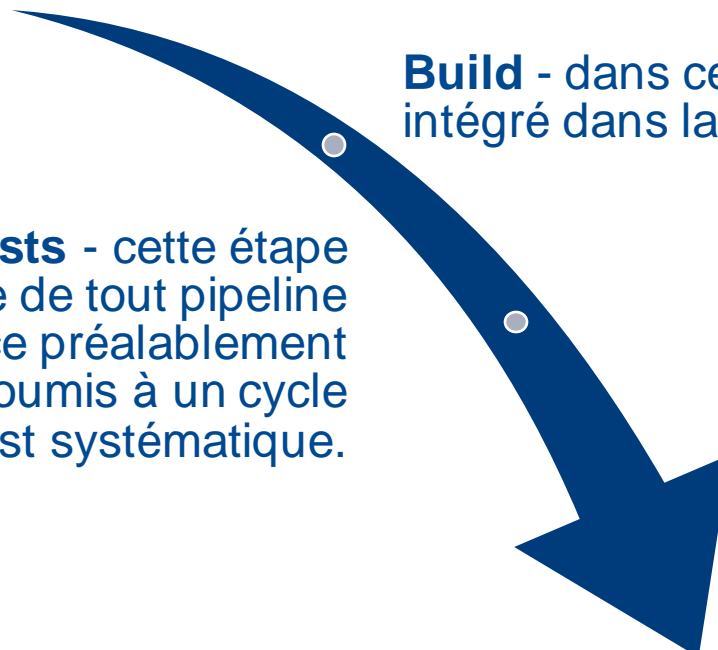
Etapes d'un pipeline CI/CD

Commit - il s'agit de la phase réelle au cours de laquelle les développeurs valident les modifications apportées au code.

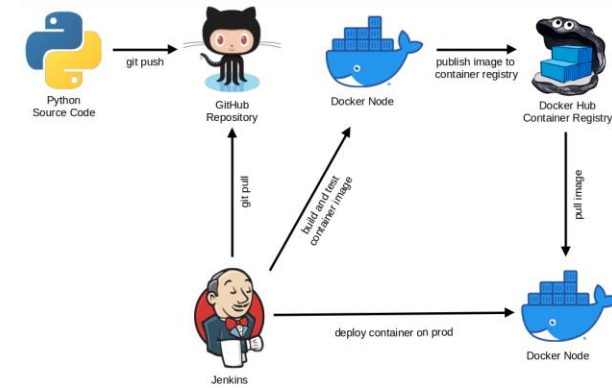
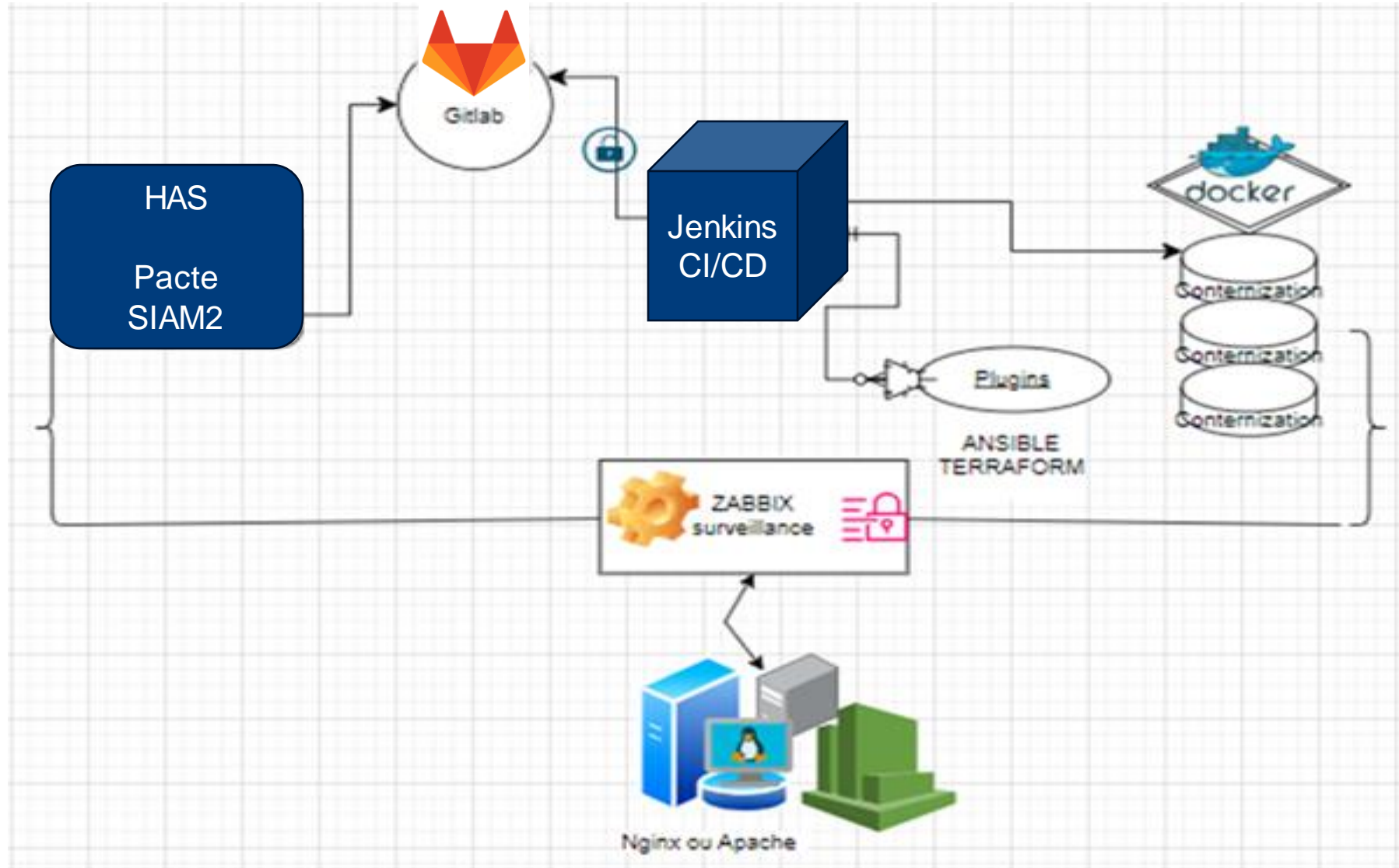
Build - dans cette phase, le code source est intégré dans la build à partir du référentiel.

Automatisation des tests - cette étape fait partie intégrante de tout pipeline CI/CD. Le code source préalablement intégré dans le build est soumis à un cycle de test systématique.



Déploiement - la version testée est finalement envoyée pour déploiement dans cette phase.



mini-schéma



Présentation de solution Pipeline CI/CD

Outil	JENKINS 	GITLAB 
Maturité	Jenkins a été lancé sous le nom de Hudson mais a été forké et renommé Jenkins en 2011.	Une plateforme plus récente créée en 2014.
PRIX	Open source	La plupart des fonctionnalités sont disponibles dans la version gratuite. L'outil a également des plans payants avec plus de fonctionnalités.
Pré-requis	Environnement d'exécution Java	Node.JS, Git, Ruby
Facilité d'utilisation	Elle peut être compliquée pour les débutants.	Facile à utiliser
Hébergement	Internal et externe	Internal et externe

Architecture de GITLAB vs JENKINS dans Pipeline CI/CD

Les deux présentent des différences distinctes qui affectent la façon dont ils gèrent le CI/CD process:



GITLAB CI

- **Intégration native** : directement dans GitLab, gestion du code et des pipelines CI/CD.
- **Pipeline as Code** : Définissez les pipelines dans le fichier `gitlab-ci.yml`
- **Facilité de configuration** : Interface flexible, conviviale pour déclencheurs des variables d'environnement.
- **Intégration continue** : Automatise le processus d'intégration pour vérifier et tester les modifications de code.
- **Déploiement continu** : automatique des applications après la validation des tests.

JENKINS



- **Personnalisation** : Grande flexibilité pour configurer et personnaliser selon les besoins spécifiques du projet et l'automatisation.
- **Support multi-plateforme** : Prise en charge de systèmes d'exploitation, langages de programmation et de déploiement.
- **Évolutivité** : Possibilité de déployer dans un environnement à grande échelle et de configurer pour s'adapter à des besoins complexes.
- **Plugins extensifs** : Vaste écosystème de plugins permettant d'étendre les fonctionnalités et répondre à divers cas d'utilisation.
- **Planifie les builds**: les tâches avec l'exécution réelle. Utilise une architecture maître-travailleur pour gérer les builds.

JENKINS Qu'est-ce que c'est?



Description

- **Approche modulaire** : Jenkins adopte une approche modulaire où chaque fonctionnalité est gérée par des plugins distincts.
- **Flexibilité et extensibilité** : écosystème riche en **plugins**, offre une flexibilité extrême pour personnaliser les pipelines CI/CD en fonction des besoins du projet.
- **Un outil d'automatisation de serveur open source:** permet d'intégrer rapidement des changements; de trouver des problèmes tôt en automatisant les processus de construction et de test.
- **Agit d'une architecture distribuée:** un serveur central exécute des "tâches" un répertoire de code source, déclenchées par divers événements: commit, build, test et déploiement par les agents.
- **Conçu pour les projets logiciels CI/CD:** devenu un pilier de l'industrie DevOps.

Configuration système requise

JENKINS

Prérequis

- Installation ou vérification que Java a été installé
- ```
apt-get install openjdk-11-jdk
```
- ```
# java -version
```

RAM

- 256 MB of RAM
 - 1 GB drive space
- Recommandé en petite/moyenne entreprise:**
- 4 GB+ of RAM
 - 50 GB+ 10 GB Docker container - drive space

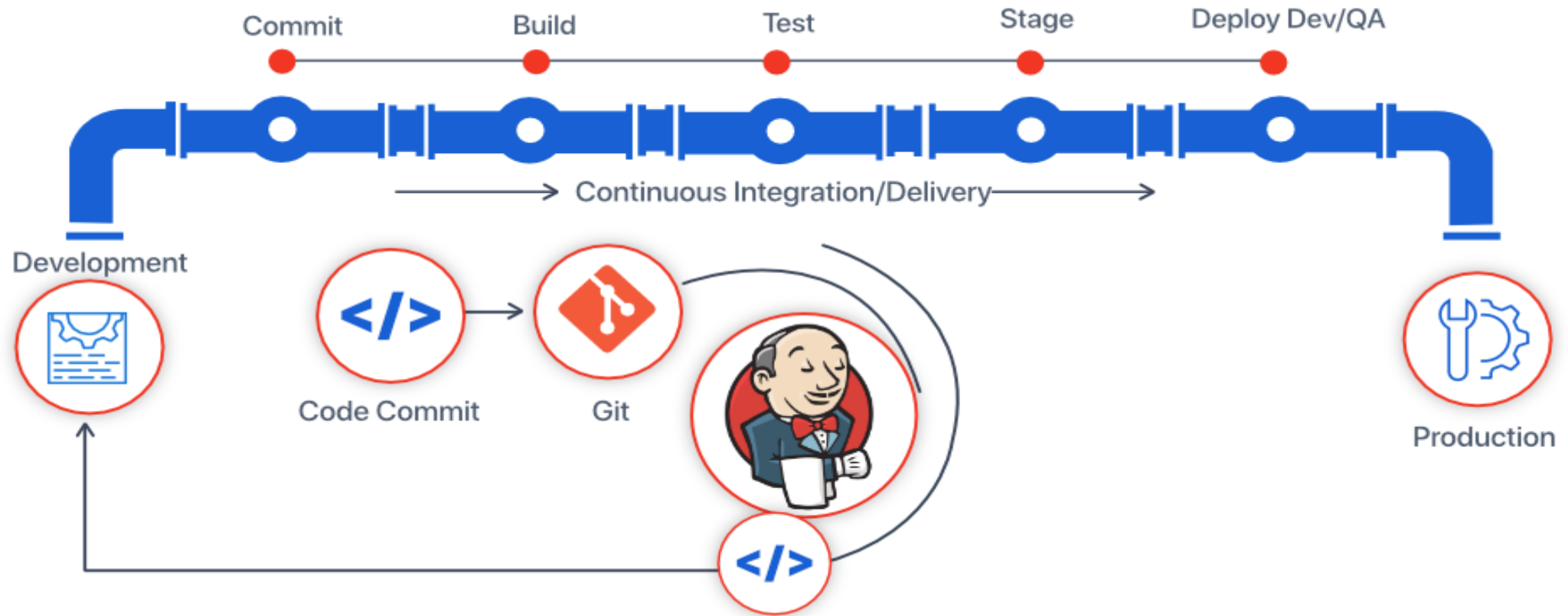
INSTALLATION de JENKINS:

```
# curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

```
# echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
```

```
# sudo apt-get update
# sudo apt-get install jenkins
# sudo systemctl start jenkins.service
```

JENKINS PIPELINE CI/CD



AVANTAGE

Sécurisée avec de clés.

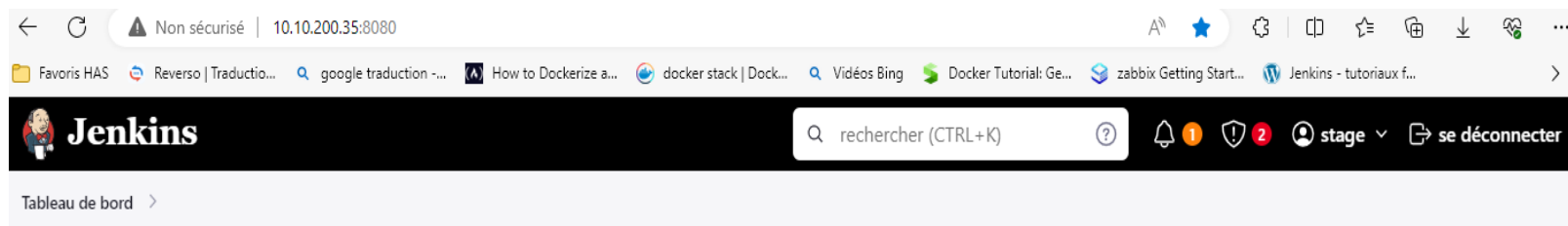


Credentials

T	P	Store ↓	Domain	ID
		Jenkins	(global)	1c
		Jenkins	(global)	43

JENKINS

Continuous Build Trigger chaque commit à Gitlab



+ Nouveau Item

Ajouter une description

Utilisateurs

Tous +

Historique des constructions

Relations entre les builds

Vérifier les empreintes numériques

Administrer Jenkins

Mes vues

File d'attente des constructions

File d'attente des constructions vide

État du lanceur de compilations

built-in node (0 of 2 executors busy)

S	M	Nom du projet	Dernier succès	Dernier échec	Dernière durée
✓	☀	gitlab_connect_jenkins	2 mn 21 s #3	s. o.	0,61 s
✓	☀	gitlab_web_jenkins	2 mn 11 s #2	s. o.	0,35 s
✓	☀	python_app	5 j 23 h #11	s. o.	0,77 s
✓	☀	stage	5 j 23 h #12	s. o.	0,52 s
✓	☀	test_docker_container	5 j 23 h #14	7 j 1 h #3	0,42 s
✓	☀	testgit	11 j #21	12 j #4	0,39 s

Icône: S M L

Légende

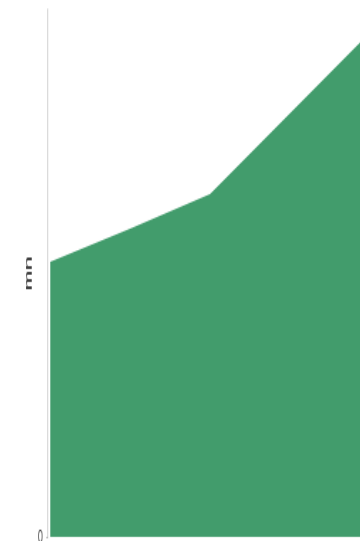
Atom feed for all

Atom feed for failures

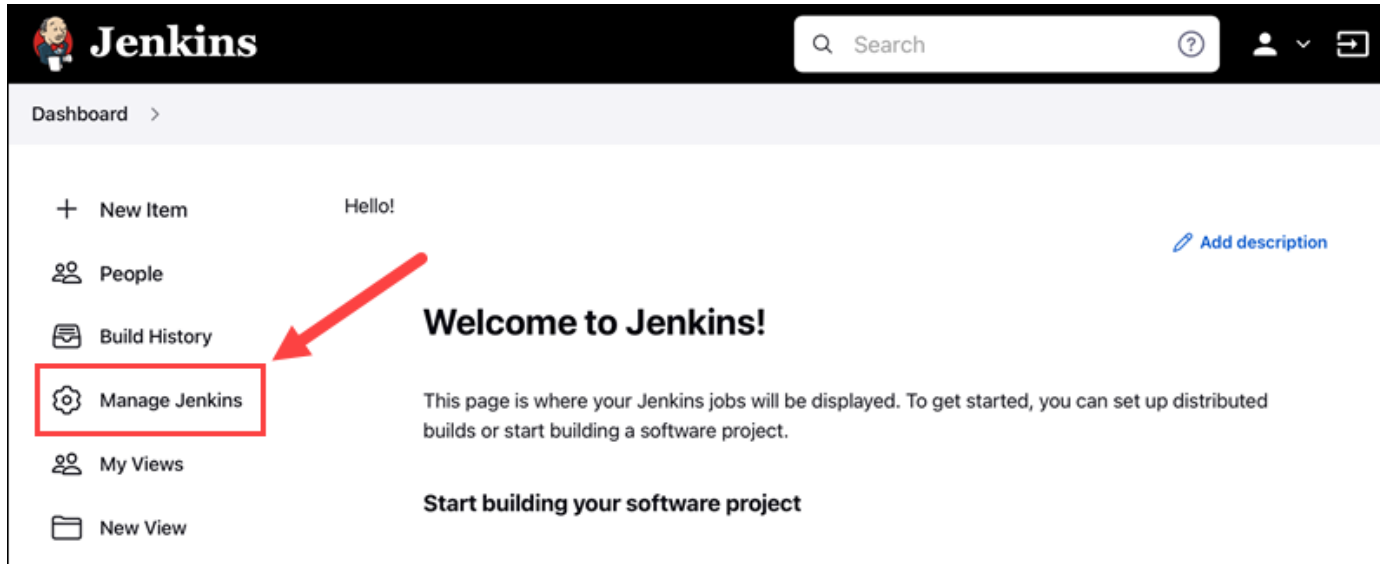
Atom feed for just latest builds

Tendance des temps de construction

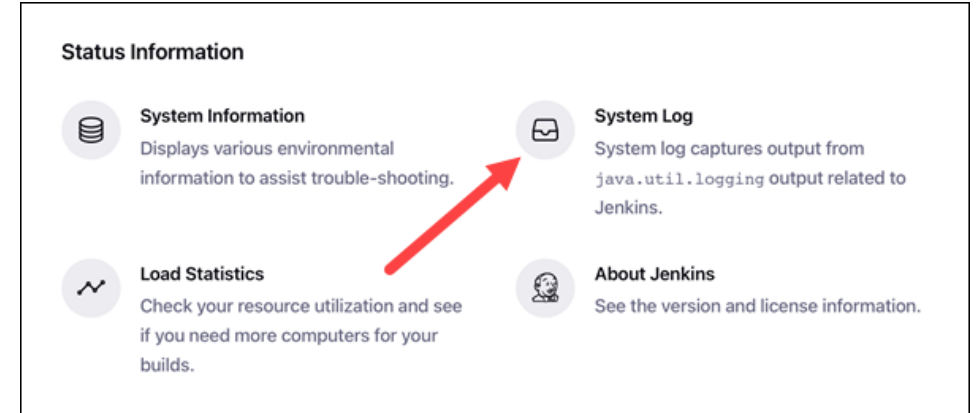
Build 1 Durée
#3 0,61 s
#2 0,42 s
#1 0,33 s



Jenkins Système Log



The screenshot shows the Jenkins Dashboard. The left sidebar contains a list of links: 'New Item', 'People', 'Build History', 'Manage Jenkins' (highlighted with a red box and a red arrow), 'My Views', and 'New View'. The main content area displays a 'Welcome to Jenkins!' message and a 'Start building your software project' button. A red arrow points from the 'Manage Jenkins' link to the 'Welcome to Jenkins!' message.



The screenshot shows the 'Status Information' section of the Jenkins interface. It contains four items: 'System Information', 'System Log' (highlighted with a red arrow), 'Load Statistics', and 'About Jenkins'. The 'System Log' item is described as 'System log captures output from java.util.logging output related to Jenkins.'

Jenkins Log

Log messages at a level lower than INFO are never recorded in the Jenkins log. Use [log recorders](#) to record these log messages.

```
Jul 19, 2022 4:32:41 PM INFO hudson.WebAppMain contextInitialized
Jenkins home directory: /Users/marko/.jenkins found at: $user.home/.jenkins
Jul 19, 2022 4:32:42 PM INFO org.eclipse.jetty.server.handler.ContextHandler doStart
Started w.@569bf9eb{Jenkins v2.359,,file:///Users/marko/.jenkins/war/,AVAILABLE}
{/Users/marko/.jenkins/war}
Jul 19, 2022 4:32:42 PM INFO org.eclipse.jetty.server.AbstractConnector doStart
Started ServerConnector@37374a5e{HTTP/1.1, (http/1.1)}{127.0.0.1:8080}
Jul 19, 2022 4:32:42 PM INFO org.eclipse.jetty.server.Server doStart
```

Console Output

```
Started by user unknown or anonymous
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\Example
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Env Variables)
[Pipeline] echo
The current build number is 1
[Pipeline] echo
Another method is to use ${BUILD_NUMBER}, which is 1
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

GITLAB Qu'est-ce que c'est?



Description

- **Approche tout-en-un** : une plateforme de développement logiciel complète qui intègre nativement les fonctionnalités de gestion de code source, de suivi des problèmes, de CI/CD, etc.
- **Dispose d'une architecture monolithique** : les fonctionnalités sont regroupées dans une seule application avec l'intégration native, offrant une expérience unifiée pour la gestion du code.
- **Pipeline as Code** : **Architecture intégrée**, définis les pipelines CI/CD à l'aide d'un fichier `gitlab-ci.yml` et propose un ensemble d'outils pour gérer du cycle de développement.
- **Intégration continue** : Automatise le processus d'intégration, vérifier et tester les modifications de code.
- **Déploiement continu** : Déploiement automatique des applications sur différentes plateformes après la validation des tests.
- **Facilité de configuration** : déclencheurs et des variables d'environnement.

Gitlab CI/CD

Création d'un RUNNER

← ↻ 🔒 https://gitlab.has-sante.fr/e.bernardez/stage-gitconnect/-/runners/new

Favoris HAS Reverso | Traductio... google traduction -... How to Dockerize a... docker stack | Dock... Vidéos Bing Docker Tutorial: Ge... zabbix Getting St

Eloisa BERNARDEZ / stage-gitconnect / CI/CD Settings / New runner

New project runner

Create a project runner to generate a command that registers the runner with all its configurations.

Platform

Operating systems

☒ Linux ☐ macOS ☐ Windows

Containers

Docker Kubernetes

Registration de Gitlab Runner


Register runner

GitLab Runner must be installed before you can register a runner. [How do I install GitLab Runner?](#)

Step 1

Copy and paste the following command into your command line to register the runner.

```
$ gitlab-runner register
--url https://gitlab.has-sante.fr
--token glrt-SusynB4bU3m75wHPkgjd
```

 The runner authentication token `glrt-SusynB4bU3m75wHPkgjd` be accessed again from the UI.

Step 2

Choose an executor when prompted by the command line. Execute

Step 3 (optional)

Manually verify that the runner is available to pick up jobs.

```
$ gitlab-runner run
```

This may not be needed if you manage your runner as a [system or user](#).

[View runners](#)

```
eloisa@toulouse: ~
Version:      16.11.1
Git revision: 535ced5f
Git branch:   16-11-stable
GO version:   go1.21.9
Built:        2024-05-03T15:52:38+0000
OS/Arch:      linux/amd64
eloisa@toulouse:~$ sudo visudo
eloisa@toulouse:~$ gitlab-runner register --url https://gitlab.has-sante.fr --
token glrt-SusynB4bU3m75wHPkgjd
Runtime platform                          arch=amd64 os=linux pid=4794
38 revision=535ced5f version=16.11.1
WARNING: Running in user-mode.
WARNING: The user-mode requires you to manually start builds processing:
WARNING: $ gitlab-runner run
WARNING: Use sudo for system-mode:
WARNING: $ sudo gitlab-runner...

Created missing unique system ID                      system_id=s_9a24f52dlf25
Enter the GitLab instance URL (for example, https://gitlab.com/):
[https://gitlab.has-sante.fr]: ^X
ERROR: Verifying runner... client error                runner=SusynB4bU status=pars
e "\x18/api/v4/": net/url: invalid control character in URL
PANIC: Failed to verify the runner.
eloisa@toulouse:~$
```

Runner de Gitlab est bien crée, enfin le Pipeline peut commencer

- Settings
- General
- Integrations
- Webhooks
- Access Tokens
- Repository
- Merge requests
- CI/CD
- Packages and registries
- Monitor

Auto DevOps

Automate building, testing, and deploying your applications based on your continuous

Runners

Runners are processes that pick up and execute CI/CD jobs for GitLab. [What is GitLab](#)

Register as many runners as you want. You can register runners as separate users, on

How do runners pick up jobs?

Runners are either:

- **active** - Available to run jobs.
- **paused** - Not available to run jobs.

Tags control which type of jobs a runner can handle. By tagging a runner, you make si

Project runners

These runners are assigned to this project.

[New project runner](#)



Assigned project runners

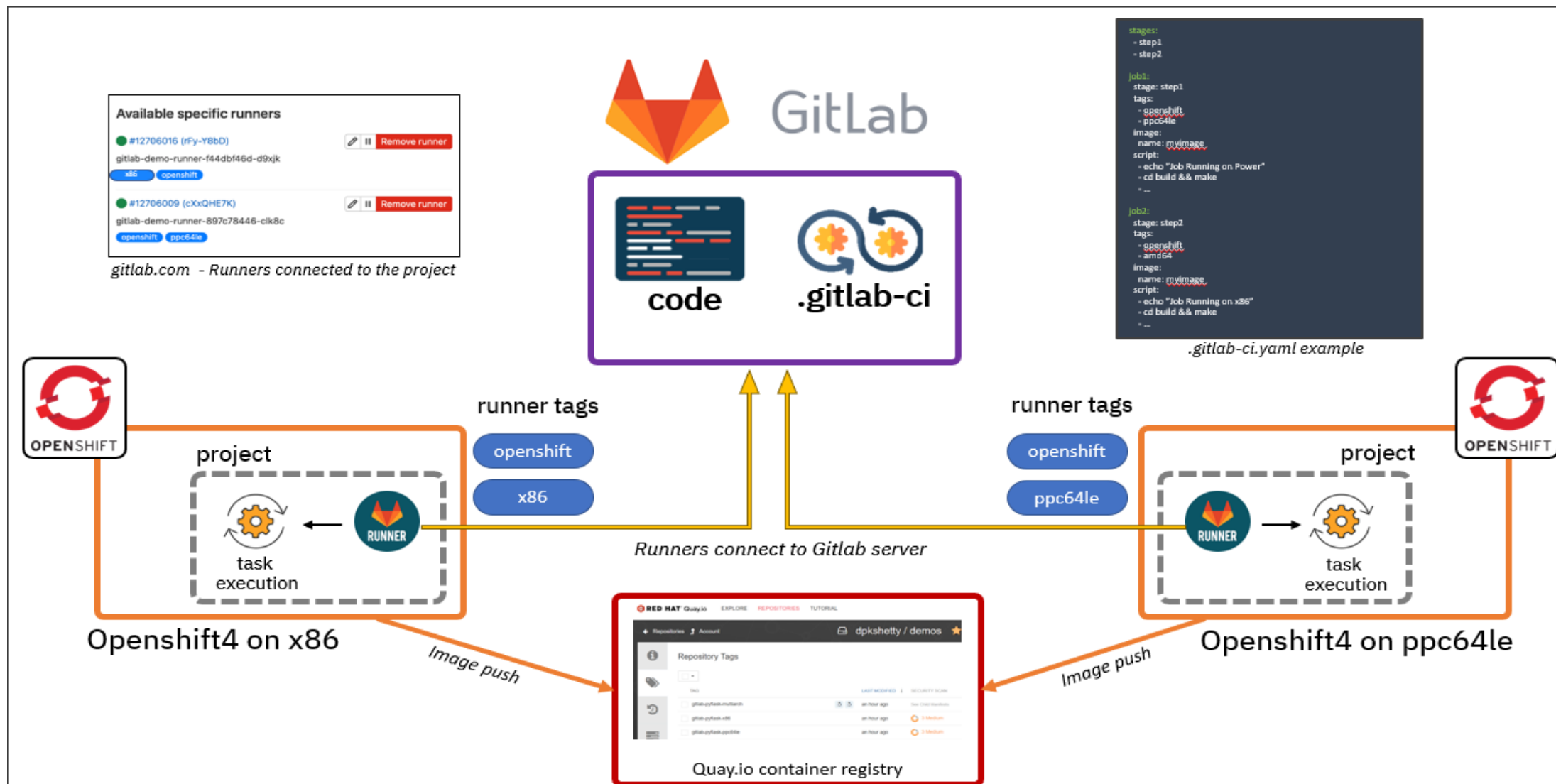
#175 (SusynB4bU)

stage-gitconnect



[Remove runner](#)

Gitlab CI/CD Pipeline

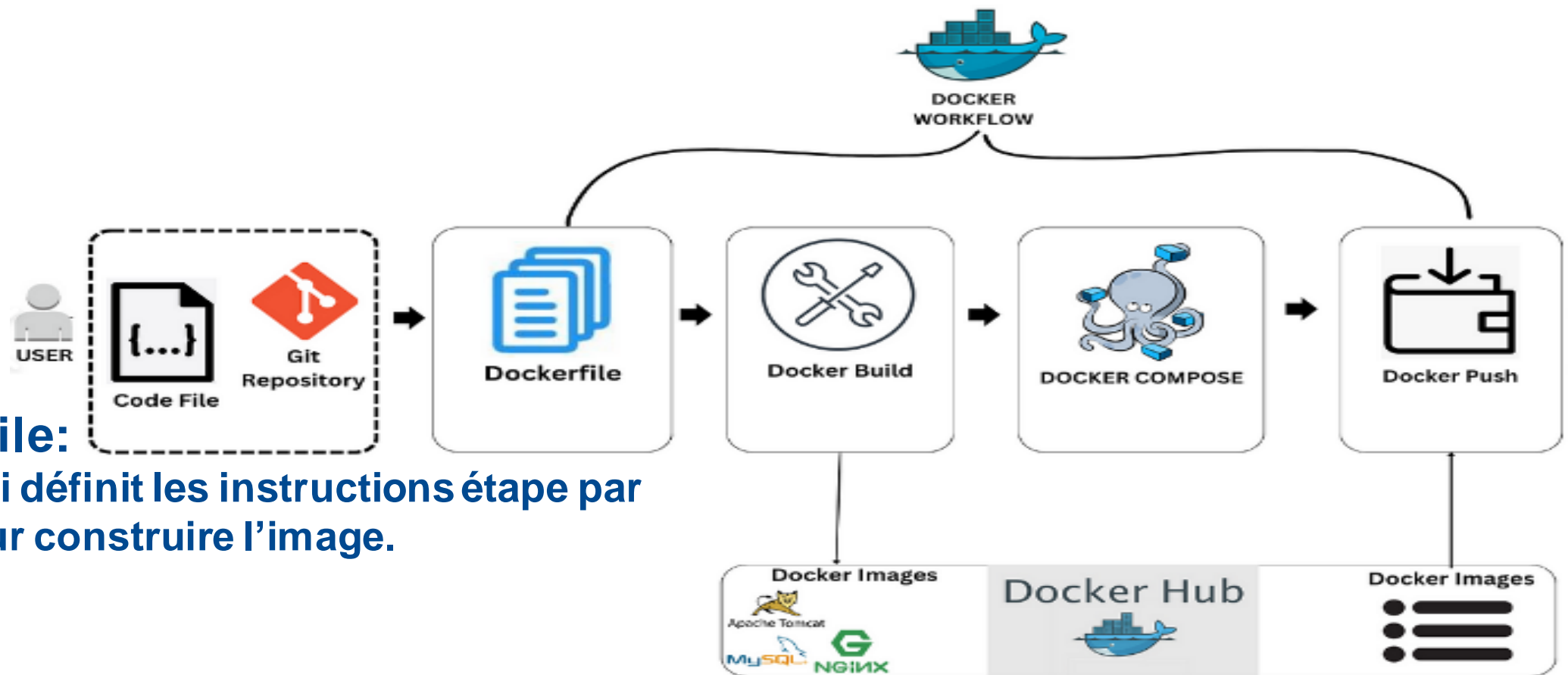


DOCKER



AVANTAGES	Description
Portabilité Stockage sans serveur	<ul style="list-style-type: none">• Permet de créer des applications portables• Déployées facilement sur n'importe quelle infrastructure.• N'exigent pas beaucoup de mémoire active pour fonctionner de manière fiable
Conteneurisation	<ul style="list-style-type: none">• En utilisant la technologie des conteneurs, les applications contiennent que les éléments nécessaires à l'exécution.• Élimination des configurations redondantes.
Déploiement rapide Légèreté Taille moyenne d'image Docker	<ul style="list-style-type: none">• L'image Docker a une taille réduite qui contient que les éléments essentiels à l'exécution d'une application.• 100 Mo à 200 Mo• inclut l'application elle-même et les bibliothèques nécessaires.

Docker WORRFLOW



Dockerfile:
fichier qui définit les instructions étape par étape pour construire l'image.

Exemple de Dockerfile



```
eloisa@toulouse: ~/stage-gitpacte/pacte
GNU nano 6.2 Dockerfile
# Use the official Nginx image as a base image
FROM nginx:latest

# Set the working directory in the container
WORKDIR /usr/share/nginx/html

# Create a personalized greeting HTML file
RUN echo "<!DOCTYPE html>" > index.html \
    && echo "<html>" >> index.html \
    && echo "<head><title>Présentation de Schéma Cible @ HAS</title></head>" >> index.html \
    && echo "<body>" >> index.html \
    && echo "<h1>Welcome to Docker PHP App</h1>" >> index.html \
    && echo "<p>Dockerfile de PACTE</p>" >> index.html \
    && echo "<p>Visitez notre site web: <a href='https://www.has-sante.fr'>www.has-sante.fr</a></p>" >> index.html \
    && echo "</body>" >> index.html \
    && echo "</html>" >> index.html

# Copy the application files into the container
COPY index.php .
COPY composer.json .
COPY phpunit.xml.dist .
COPY src ./src
COPY webroot ./webroot
COPY bin ./bin
COPY config ./config
COPY lib ./lib
COPY tests ./tests
COPY vendor ./vendor
COPY README.md .

# Expose port 8008 for the application
EXPOSE 8010
```

Virtualisation

Préparation d'une Machine Virtuelle (VM) avec HYPERVISEUR

Crée une machine virtuelle

Virtual machine Name and Operating System


Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Nom : ✓

Folder: ▼


ISO Image: ▼

Edition: ▼

Type : ▼ 

Version : ▼

☐ Skip Unattended Installation

 No ISO image is selected, the guest OS will need to be installed manually.

Aide Mode expert Précédent **Suivant** Annuler

Installation d'un OS	<ul style="list-style-type: none">Linux (Ubuntu)Windows
Installation des outils DevOps:	<ul style="list-style-type: none">DockerJenkinsAnsibleTerraformZabbix

Virtualisation

Machine Virtuelle (VM)

The screenshot displays the Oracle VM VirtualBox - Gestionnaire de machines window. The interface is in French and shows a list of virtual machines on the left, with 'serverzabbix' selected. The main panel shows the configuration for 'serverzabbix' across several tabs: Général, System, Affichage, Stockage, Audio, and Réseau. The 'Prévisualisation' tab on the right shows a preview of the virtual machine's display, which is currently black with the text 'serverzabbix' in white.

Oracle VM VirtualBox - Gestionnaire de machines

Fichier Machine Aide

Outils

serverzabbix Éteinte

Général

Nom : serverzabbix
Système d'exploitation : Ubuntu (64-bit)

System

Mémoire vive : 4096 Mo
Processeurs : 2
Ordre d'amorçage : Disque dur, Optique, Disquette
Accélération : Pagination imbriquée, Paravirtualisation KVM

Affichage

Mémoire vidéo : 16 Mo
Contrôleur graphique : VMSVGA
Serveur de bureau à distance : Désactivé
Enregistrement : Désactivé

Stockage

Contrôleur : IDE
Maître secondaire IDE : [Lecteur optique] Vide
Contrôleur : SATA
Port SATA 0 : serverzabbix.vdi (Normal, 25,00 Gio)

Audio

Pilote hôte : Par défaut
Contrôleur : ICH AC97

Réseau

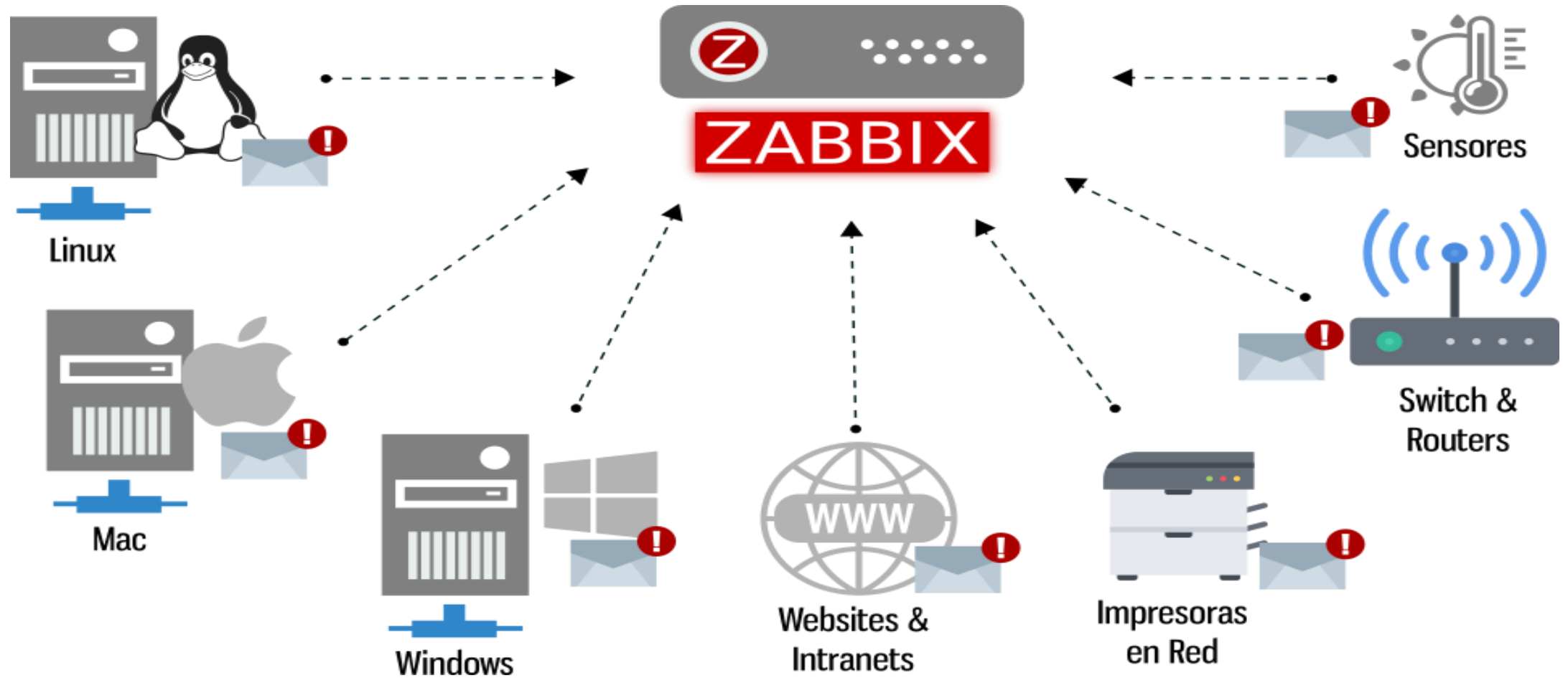
Interface 1: Intel PRO/1000 MT Desktop (Interface pont Realtek RTL8852AE WiFi 6 802.11ax PCIe Adapter)

Prévisualisation

serverzabbix

Proposition de monitoring

logiciel qui supervise de nombreux paramètres réseaux:



Monitoring

En fonction] - Oracle VM VirtualBox

Écran Entrée Périphériques Aide

Activities Firefox Web Browser

mai 6 14:42

Zabbix

The screenshot shows the Zabbix monitoring dashboard running in a Firefox web browser within an Oracle VM VirtualBox. The browser address bar shows the URL `localhost/zabbix/zabbix.php?action=dashboard.view`. The dashboard interface includes a left sidebar with navigation links for Dashboards, Monitoring, Services, Inventory, Reports, Data collection, Alerts, Users, Administration, Support, Integrations, Help, and User settings. The main content area displays several widgets: 'Top hosts by CPU utilization' showing the Zabbix server at 34.17% utilization; a large '2.41' value with a downward arrow indicating 'Zabbix server Values per sec...'; 'System information' showing Zabbix server is running, version 7.0.0beta3, and 213 items supported; 'Host availability' showing 2 available hosts; 'Problems by severity' showing 2 warning problems; and 'Current problems' listing two recent restarts of the ubuntu-server-stage and Zabbix server.

ZABBIX

zabbix

Search

Dashboards

- Monitoring
- Services
- Inventory
- Reports
- Data collection
- Alerts
- Users
- Administration
- Support
- Integrations
- Help
- User settings

Show Applications

All dashboards / Global view

Top hosts by CPU utilization

Host name	Utilization	1m avg	5m avg	15m avg	Pro
Zabbix server		34.17 %	2.64	1.10	0.43

2.41 ↓

Zabbix server Values per sec...

System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Zabbix server version	7.0.0beta3	
Zabbix frontend version	7.0.0beta3	
Number of hosts (enabled/disabled)	2	2 / 0
Number of templates	310	
Number of items (enabled/disabled/not supported)	213	202 / 0 / 11

Host availability

2 Available	0 Not available	0 Mixed	0 Unknown	2 Tot
-------------	-----------------	---------	-----------	-------

Problems by severity

0 Disaster	0 High	0 Average	2 Warning	0 Information	0 Not classified
------------	--------	-----------	-----------	---------------	------------------

Current problems

Time	Info	Host	Problem • Severity	Duration	Update	Actions	Tags
12:41:02 PM		ubuntu-server-stage	Linux: ubuntu-server-stage has been restarted (uptime < 10m)	1m 38s	Update		class: os component: system scope: notice ...
12:41:00 PM		Zabbix server	Linux: Zabbix server has been restarted (uptime < 10m)	1m 40s	Update		class: os component: system scope: notice ...

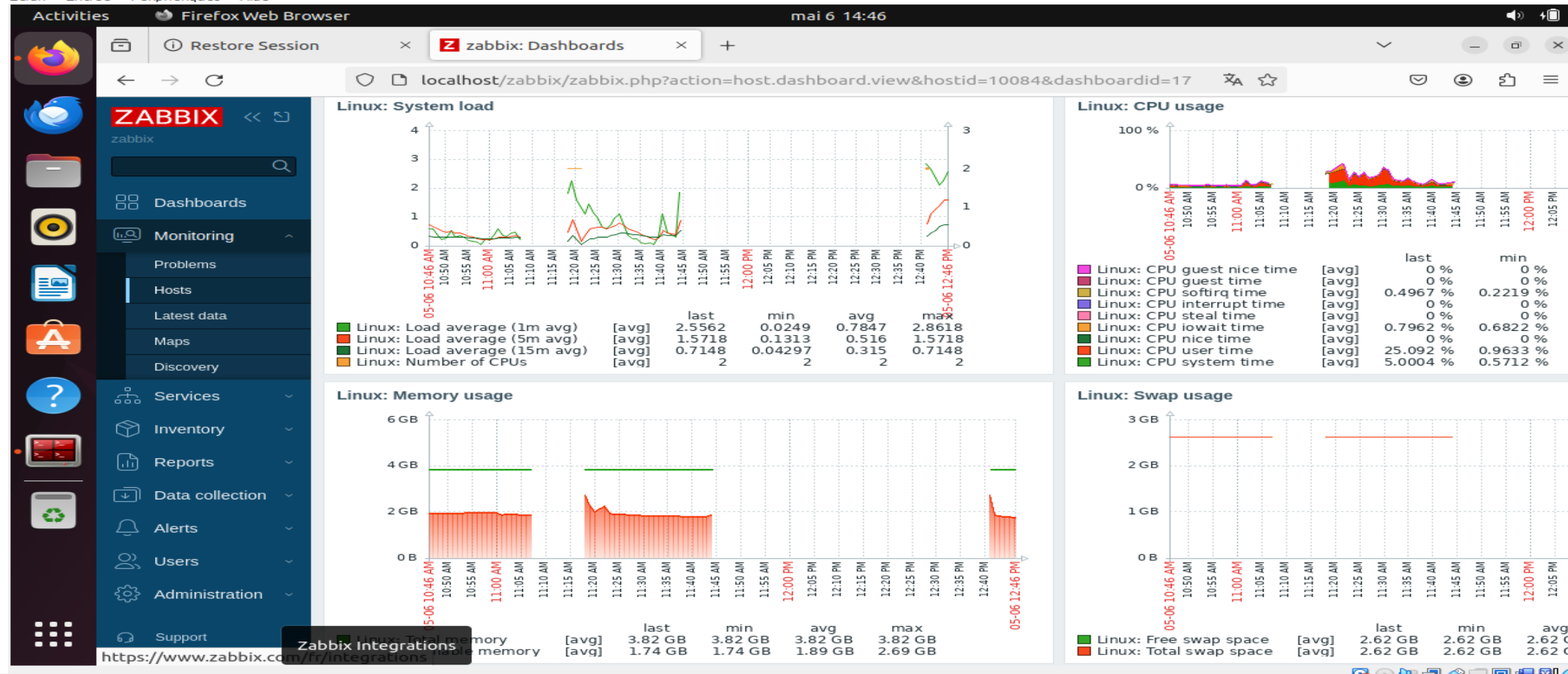
Geomap

+
-
Home

Zabbix Dashboard

fonction] - Oracle VM VirtualBox

Écran Entrée Périphériques Aide





2

SIEM

Plan

1. Enjeux
2. Exploration de la solution
3. Moyens et Ressources



Le besoin

Proposition de solutions technos (pourquoi, les +, les -, plusieurs possibles, pourquoi ce choix)

Les enjeux

Mise en place d'un System Information Events Management

Collecter les logs de deux applications de la HAS : SIAM et Pacte

Analyser les événements pour détecter les menaces et alerter les administrateurs

Process de mise en œuvre

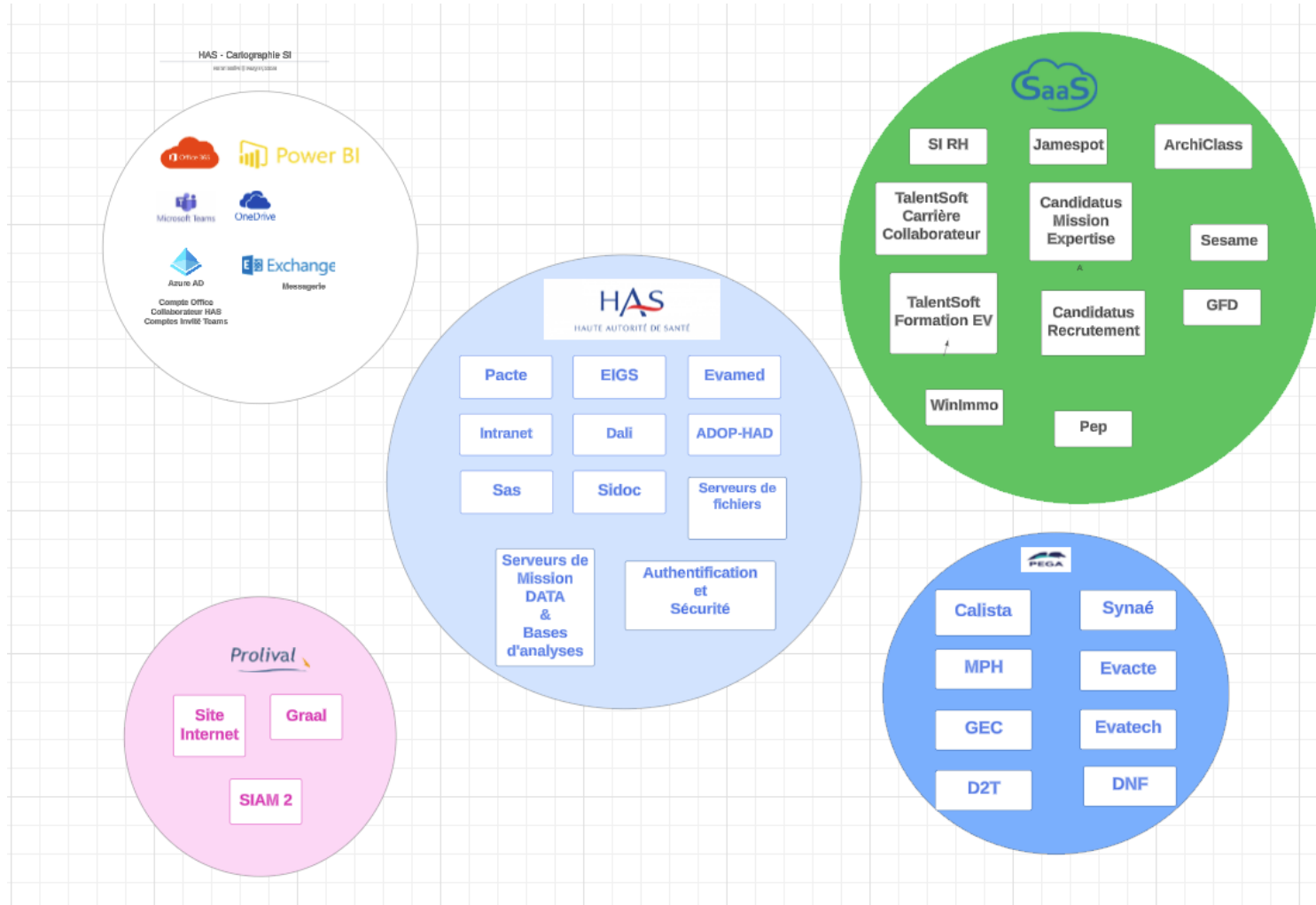
Découverte de l'architecture applicative

Choix de l'outil

Définition de l'implémentation

Planning de mise en œuvre

L'architecture applicative : compréhension et analyse



Prérequis techniques identifiés

SIAM HAS

Services

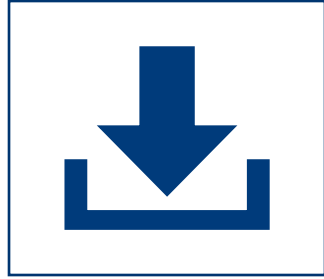
- mysqld
- jboss-eap
- slapd (LDAP)
- Alfresco
- Openoffice

PACTE

Services

- MariaDB
- php

Logiciel propriétaire ou libre ?



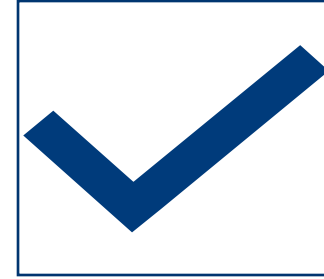
Open-source

Avantages :

- Coût: téléchargement gratuit, utilisation gratuite
- Personnalisation : liberté d'adaptation du logiciel à un environnement
- Communauté et support (conseils, solutions aux problèmes rencontrés.
- Intégration avec des outils tiers

Inconvénients:

- Support limité : en fonction de la taille et de l'activité de la communauté
- Complexité de mise en œuvre : peut nécessiter une expertise technique
- Stabilité et maturité
- Code peut contenir un backdoor



Propriétaire

Avantages

- Support dédié
- Intégration simplifiée avec d'autres produits du même fournisseur
- Fonctionnalités avancées (options de personnalisation spécifique)
- Moins risqué face aux hackers

Inconvénients :

- Coût : achat de licences et de services supplémentaires
- Dépendance au fournisseur : Mise à jour, support et évolution du produit
- Limitation de la personnalisation (restriction de propriété intellectuelle)

Le choix entre un logiciel propriétaire et open source dépend de facteurs tels que les besoins spécifiques en fonctionnalités, le budget, les compétences techniques disponibles et la tolérance au risque.



Description de 5 solutions SIEM Open Source

Outil	Description
AlienVault® OSSIM™	Le plus ancien SIEM géré par AT&T, lancé en raison du manque de produits open source disponibles. N'offre pas la gestion de logs.
OSSEC	Pour surveiller et contrôler les systèmes, il mélange tous les aspects du HIDS (détection d'intrusion basée sur l'hôte), de la surveillance des journaux
Security Onion	Fournit une solution complète et intégrée pour surveiller, analyser et défendre les réseaux informatiques contre les menaces de sécurité.
Snort	Analyser le trafic réseau, identifie et bloque les menaces de sécurité potentielles et utilise une série de règles qui aident à définir une activité réseau malveillante.
Wazuh	Fork libre du projet OSSEC, offre à la fois les fonctionnalités XDR (Extended Detection and Response) and SIEM (Security Information and Event Management) sur les assets et le cloud.



wazuh.

Les +/- des solutions (critères de sélection)

	AlienVault® OSSIM™	OSSEC+	Security Onion	Snort	Wazuh
Installation	Image ISO 1 serveur uniquement		Image Machine Iso/Cloud		1 ou +++ serveurs Cloud (Saas), Image machine OVA/Amazon, conteneurs (Docker/Kubernetes)
Gestion de logs	☒	☑	☑	☑ (réseau uniquement)	☑
Version payante	USM Anywhere™	Atomic OSSEC	☒ (sauf support et formations)		☒ (sauf support, services et formations)
Communauté	Forum			Email	
Intégration outils tiers	☒	☒	☑		☑
Simplicité de la documentation	☒ Sauf versions payantes	<u>Incomplet</u>	<u>Structuré et fournie</u>	pdf	<u>Structuré et fournie</u>
Détection de vulnérabilités	☑	☑	☒		☑
Conformité RGPD	☒	☑	☒		☑
Console web	☑	☒	☑		☑



Choix, Composants,
Configuration et Architecture

Exploration Approfondie de la Solution Wazuh

La solution Wazuh ?

Créée en 2015, Wazuh est une solution EDR open source, disponible gratuitement et complète.



SIEM

- Centralisation de logs
- Détection de vulnérabilités
- SCA (Security Configuration Assessment) :
 - tester la conformité de la configuration des systèmes – Center Of Internet Security
- Vérifier les frameworks tels RGPD, TSC SOC2, PCI DSS, NIST 800-53, et HIPAA
- Alertes et notifications par mail
- Dashboard : Les remontées, personnalisations

XDR : Détection et Réponse

- Threat hunting :
 - analyse focus détection et réponse aux événements qui se produiront
- Behavioral analysis :
 - chasse aux comportements non conforme ou suspects
- Automated response : suggestions
- Cloud, container, kubernetes
- Threat intelligence :
 - OSINT & Open data



La solution Wazuh ?



Créée en 2015, Wazuh est une solution XDR open source, disponible gratuitement et complète.



SIEM

Centralisation de logs

Détection de vulnérabilités

SCA (Security Configuration Assessment) :

- tester la conformité de la configuration des systèmes – Center Of Internet Security

Vérifier les frameworks tels RGPD, TSC
SOC2, PCI DSS, NIST 800-53, et HIPAA

Alertes et notifications par mail

Dashbord : Les remontées,
personnalisations



XDR : Détection et Réponse

Threat hunting :

- analyse focus détection et réponse aux événements qui se produiront

Behavioral analysis :

- chasse aux comportements non conforme ou suspects

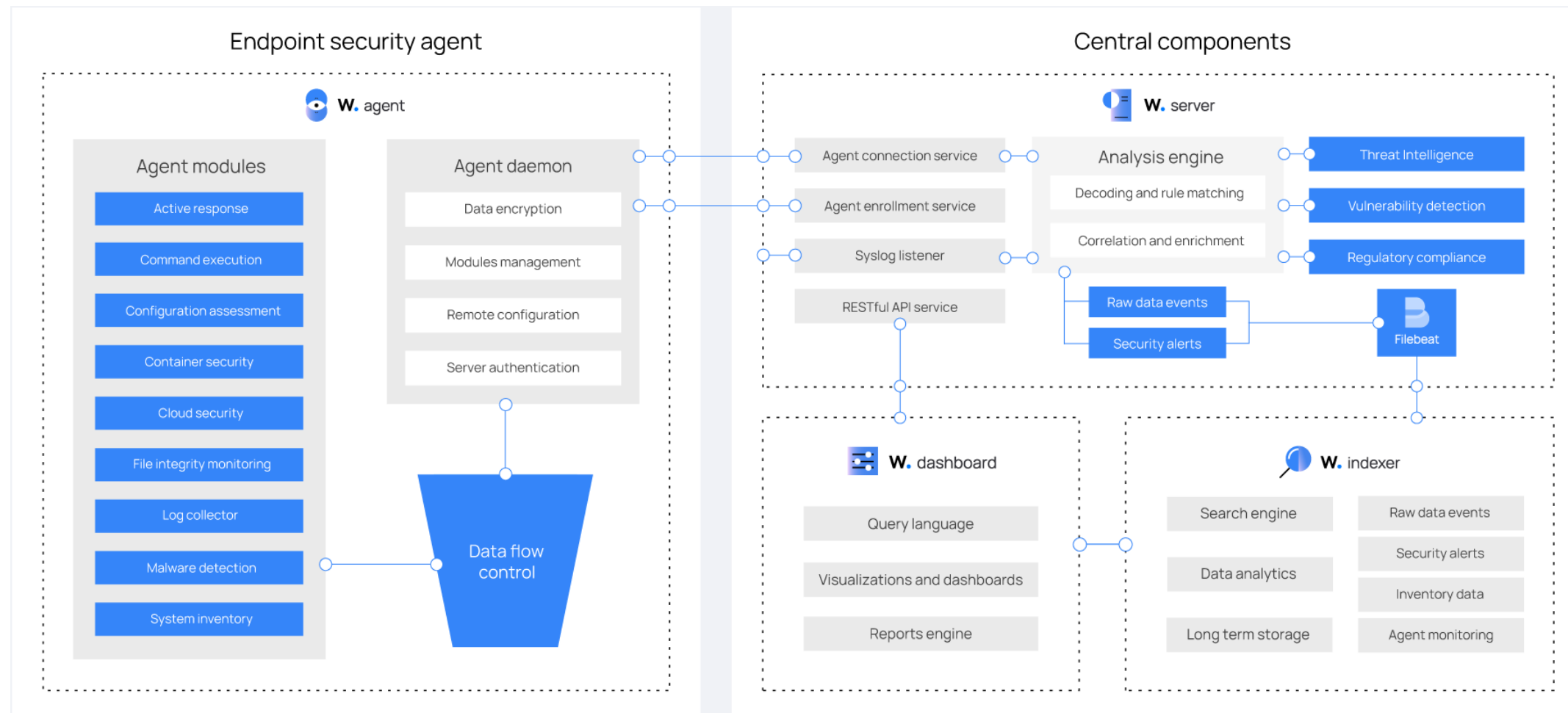
Automated response : suggestions

Cloud, container, kubernetes

Threat intelligence :

- OSINT & Open data

La plateforme Wazuh

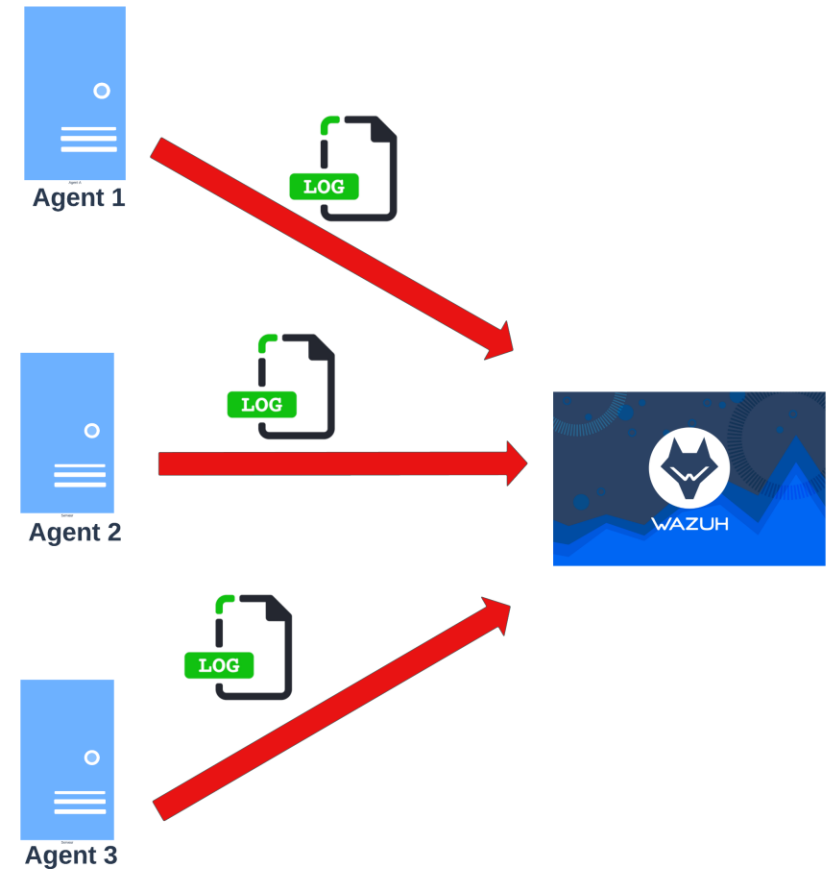


Wazuh Agents

Les agents Wazuh sont installés sur les assets tels que :

- des ordinateurs portables,
- des ordinateurs de bureau,
- des serveurs,
- des instances cloud
- ou des machines virtuelles.

Ils offrent des capacités de prévention, de détection et de réponse aux menaces.



Composants centraux Wazuh

W.server W.indexer W.dashboard

Collecter les informations des agents (roles master/worker)

Gère les agents, les configure et les met à jour à distance si nécessaire. Ce composant analyse les données reçues des agents, les traite via des décodeurs et des règles et utilise les renseignements sur les menaces pour rechercher des indicateurs de compromission.

analyse les données reçues des agents et les traite à l'aide de renseignements sur les menaces.

Un seul serveur peut analyser les données de milliers d'agents et évoluer lorsqu'il est configuré en cluster.

Moteur de recherche et d'analyse de texte

Il est chargé d'indexer et de stocker les alertes générées par le serveur Wazuh.

analyse les données reçues des agents et les traite à l'aide de renseignements sur les menaces.

Un seul serveur peut analyser les données de milliers d'agents et évoluer lorsqu'il est configuré en cluster.

Interface graphique

Une interface Web flexible et intuitive pour l'exploration, l'analyse et la visualisation de données. Le tableau de bord permet de gérer la configuration Wazuh et de surveiller son statut.

l'interface utilisateur Web pour la visualisation, l'analyse et la gestion des données.

Il comprend des tableaux de bord pour la conformité réglementaire, les vulnérabilités, l'intégrité des fichiers, l'évaluation de la configuration et les événements de l'infrastructure cloud, entre autres.

Wazuh Server & Wazuh Indexer & Dashbord

Systemes d'exploitation

64-bit Linux

Ubuntu 22.04 LTS

Processeur

Pour 1 nœud

Minimum

Recommandé

RAM (GB)

8

16

CPU (Cores)

4

8

Espace disque estimé en fonction du nombre d'alertes par secondes

Points de terminaison surveillés

APS

Stockage (GB/90 jours)

Serveurs

0,25

3,7

Stockage

500Go pour la production et 250 pour le test

Prend en charge les navigateurs Web suivants

Brave (possibilité d'activer VPN, de passer en mode TOR)

Firefox 93 ou version ultérieure

Edge devrait fonctionner puisque basé sur Chromium

Wazuh Agent

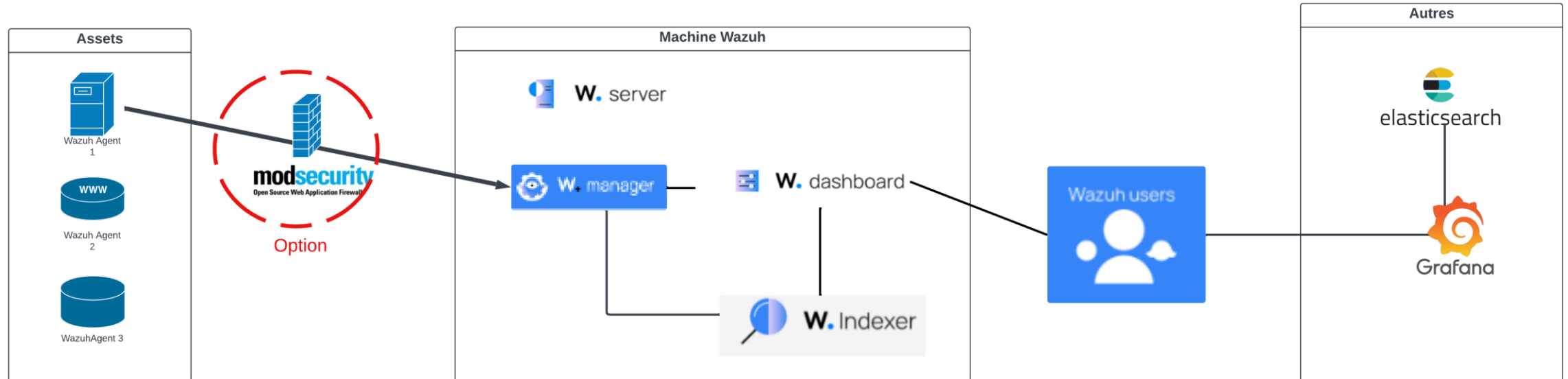
Fournit des fonctionnalités clés pour améliorer la sécurité des systèmes.

- Collecteur de journaux
- Exécution des commandes
- Surveillance de l'intégrité des fichiers (FIM)
- Évaluation de la configuration de sécurité (SCA)
- Inventaire du système
- Détection des logiciels malveillants
- Réponse active
- Sécurité des conteneurs
- Sécurité du cloud

RAM

35 MB en moyenne

Implémentation



[Cette photo](#) par Auteur inconnu est soumise à la licence [CC BY-SA](#)



[Cette photo](#) par Auteur inconnu est soumise à la licence [CC BY-SA-NC](#)



[Cette photo](#) par Auteur inconnu est soumise à la licence [CC BY-SA](#)



&

modsecurity
Open Source Web Application Firewall

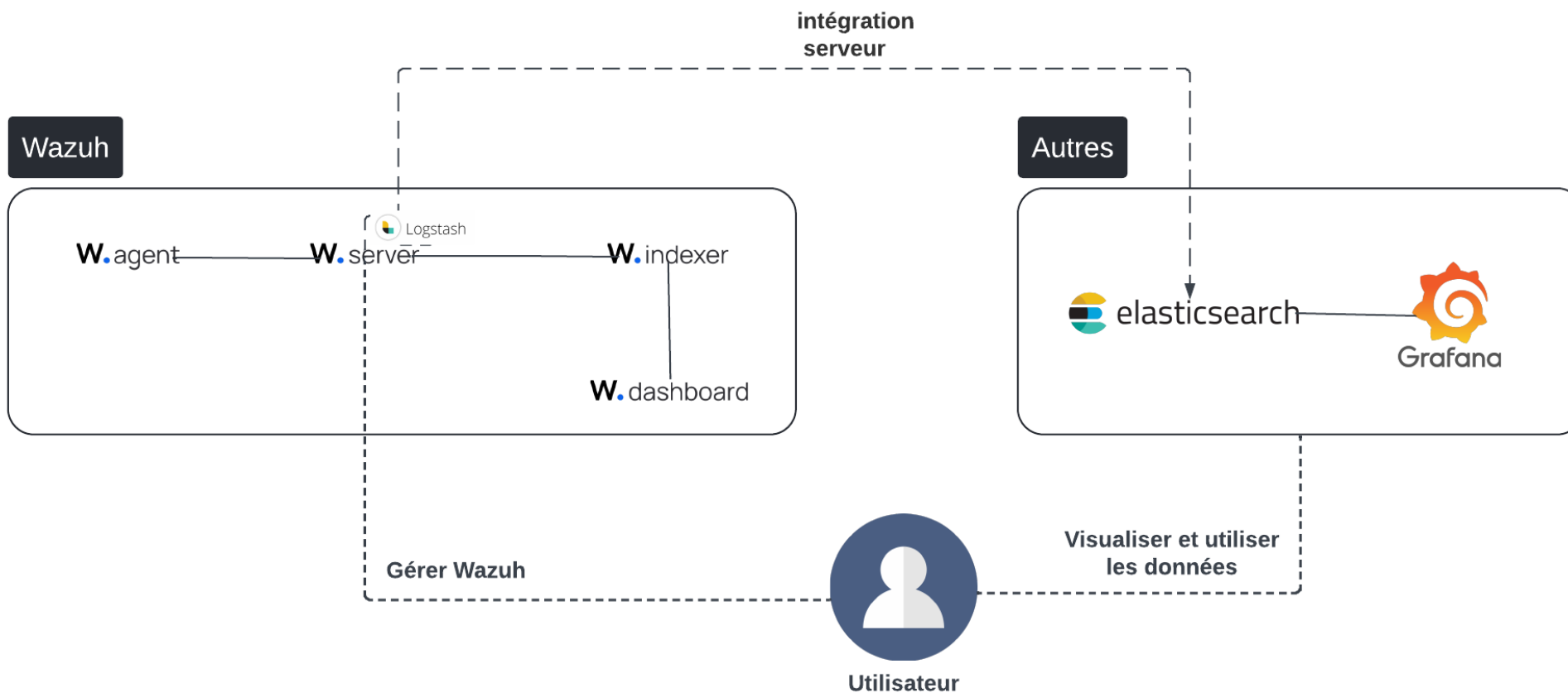
ModSecurity

- Moteur de pare-feu d'application Web (WAF) multiplateforme open source pour Apache, IIS et Nginx.
- Offre une protection contre une gamme d'attaques contre les applications Web
- permet la surveillance, la journalisation et l'analyse du trafic HTTP en temps réel.

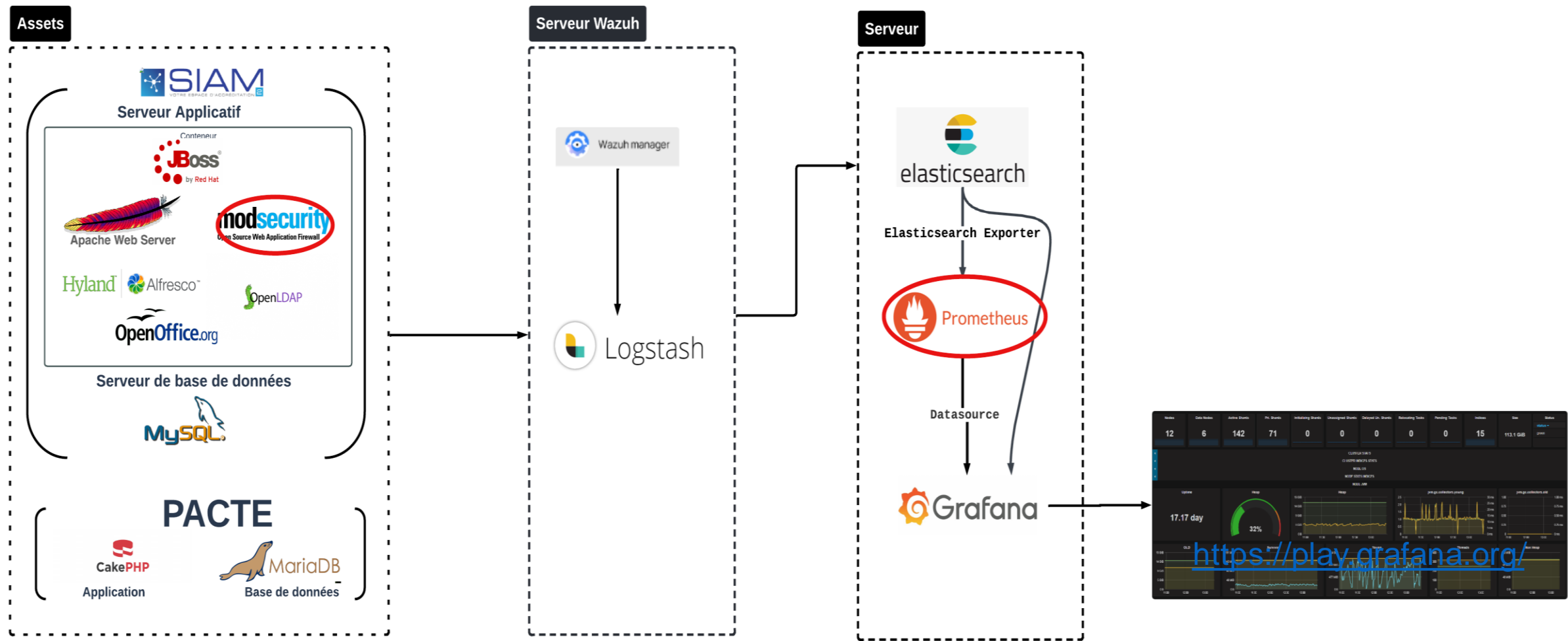
L'OWASP ModSecurity Core Rule Set (CRS)

- Ensemble de règles génériques de détection d'attaques à utiliser avec ModSecurity ou les pare-feux d'applications Web compatibles.
- Le CRS vise à protéger les applications Web contre un large éventail d'attaques.

Proposition d'architecture de déploiement



Zoom sur SIAM et Pacte





Les livrables

Le planning

Les moyens et les ressources

Les livrables

- Liste des fonctionnalités de détection des menaces, les intégrations avec d'autres systèmes de sécurité, les capacités de reporting, etc.

Spécifications fonctionnelles



- Schéma architectural, Spécifications techniques pour l'installation et la configuration du SIEM

Conception de l'architecture SIEM



- Guides d'installation, des manuels d'utilisation, etc.
- Procédures de configuration

Documentation



Planning : Estimation pour la mise en œuvre

			mai 13, 2024					mai 20, 2024					mai 27, 2024					juin 3, 2024					juin 10, 2024					juin 17, 2024					juin 24, 2024					juil 1, 2024					juil 8, 2024					juil 15, 2024									
TASK	ASSIGNED	PROGRESS	13	14	15	16	17	20	21	22	23	24	#	28	29	30	31	3	4	5	6	7	10	11	12	13	14	17	18	19	20	21	24	25	26	27	28	1	2	3	4	5	8	9	10	11	12	15	16	17	18	19					
			l	m	m	j	v	l	m	m	j	v	l	m	m	j	v	l	m	m	j	v	l	m	m	j	v	l	m	m	j	v	l	m	m	j	v	l	m	m	j	v	l	m	m	j	v										
Installation & Configuration																																																									
Pré-requis : mise à disposition serveurs	Thierry	100%																																																							
Installation des composants de Wazuh	Fadi	0%																																																							
Installation des Agents pour SIAM	SIAM	0%																																																							
Installation des Agents pour Pacte	PACTE	0%																																																							
Activation de fonctionnalités	Fadi	0%																																																							
Installation et configuration ModSecurity	SIAM	0%																																																							
Documentation Installation et configurations	Fadi	0%																																																							
Test des fonctionnalités de Wazuh																																																									
Execution des tests		0%																																																							
Visualisation																																																									
Installation de ElasticSearch	Fadi	0%																																																							
Installation de Grafana	Fadi	0%																																																							
Création de graphiques	Fadi	0%																																																							
Configuration de Grafana	Fadi	0%																																																							

Les dates, les tâches et les ressources mentionnées dans ce planning sont sujets à des ajustements au fur et à mesure que le projet progresse. Nous accueillons toute suggestion ou modification pour améliorer notre planification.

Critères de réussite

Objectifs de sécurité atteints

- Consolidation des logs multi-sources
 - Si ok, ajouter d'autres sources
 - Si ko, tester une autre solution
- Intégration dans l'infrastructure existante
- Performance et disponibilité (gestion du volume de données)
- Mesure de la valeur ajoutée
 - temps de détection des incidents
 - nombre d'incidents évités
 - Économies de coûts réalisées

Implication des différents acteurs

- Engagement
- Participation active
- Communication ouverte et transparente
- Respect des échéances et des livrables
- Adaptabilité et flexibilité

www.has-sante.fr

