

# Documentation Complète - Projet de Monitoring avec Grafana, Prometheus et Node Exporter

Une solution complète de surveillance en temps réel pour vos serveurs AWS EC2, déployée automatiquement avec Ansible.

# Table des Matières

01

---

## Aperçu du Projet

Introduction et objectifs de la stack de monitoring

02

---

## Architecture du Système

Flux de données et composants principaux

03

---

## Structure du Projet

Organisation des fichiers et rôles Ansible

04

---

## Prérequis et Installation

Configuration système et dépendances requises

05

---

## Configuration détaillée

Paramètres Ansible, inventaire et variables

06

---

## Déploiement

Processus complet de mise en production

07

---

## Accès aux Services

URLs, identifiants et dashboards disponibles

# Aperçu du Projet

Ce projet déploie une stack complète de monitoring sur des serveurs AWS EC2 en utilisant Ansible. La stack comprend :

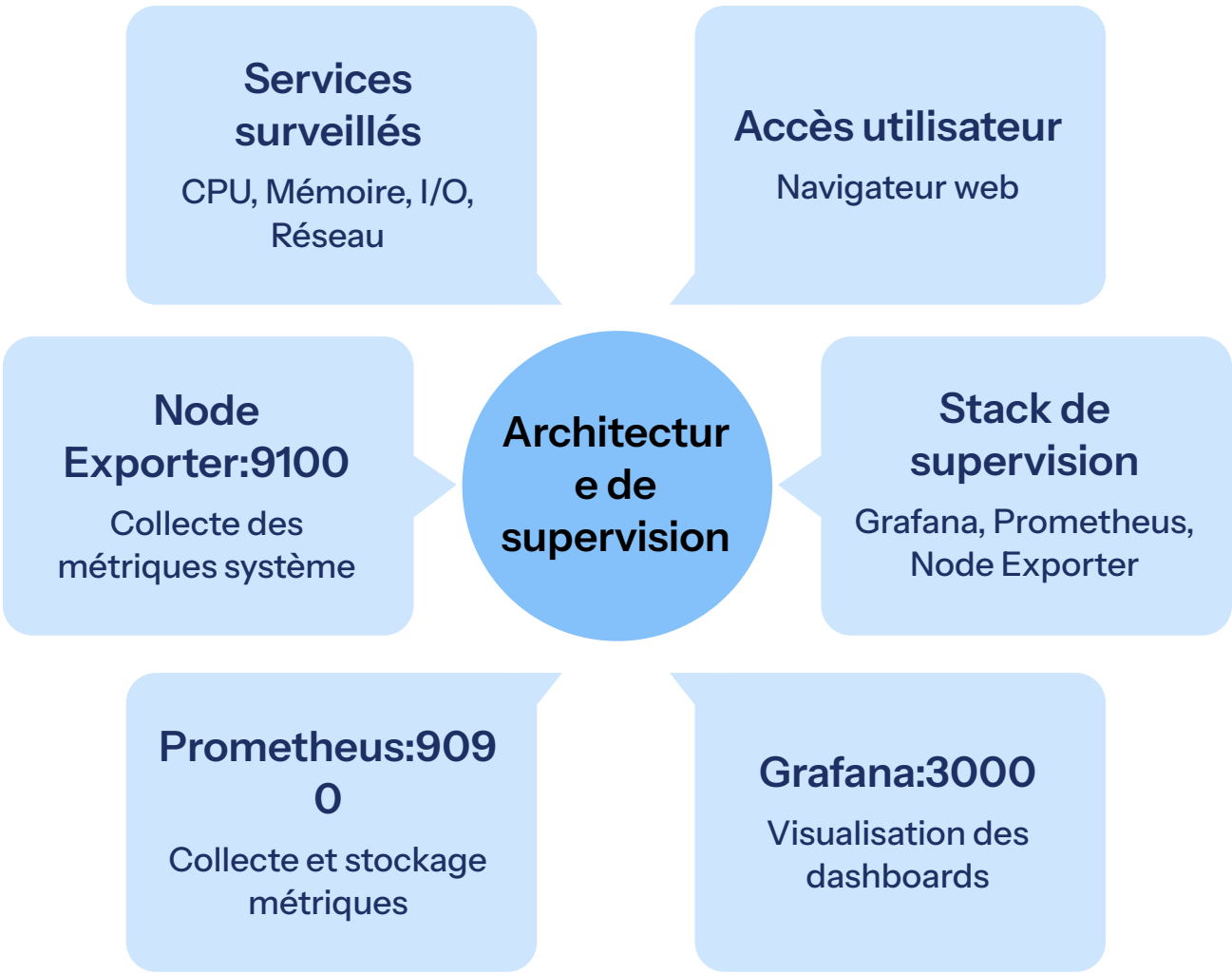
- **Grafana** : Plateforme de visualisation et de dashboard
- **Prometheus** : Système de monitoring et d>alerting
- **Node Exporter** : Collecteur de métriques système

## Objectifs

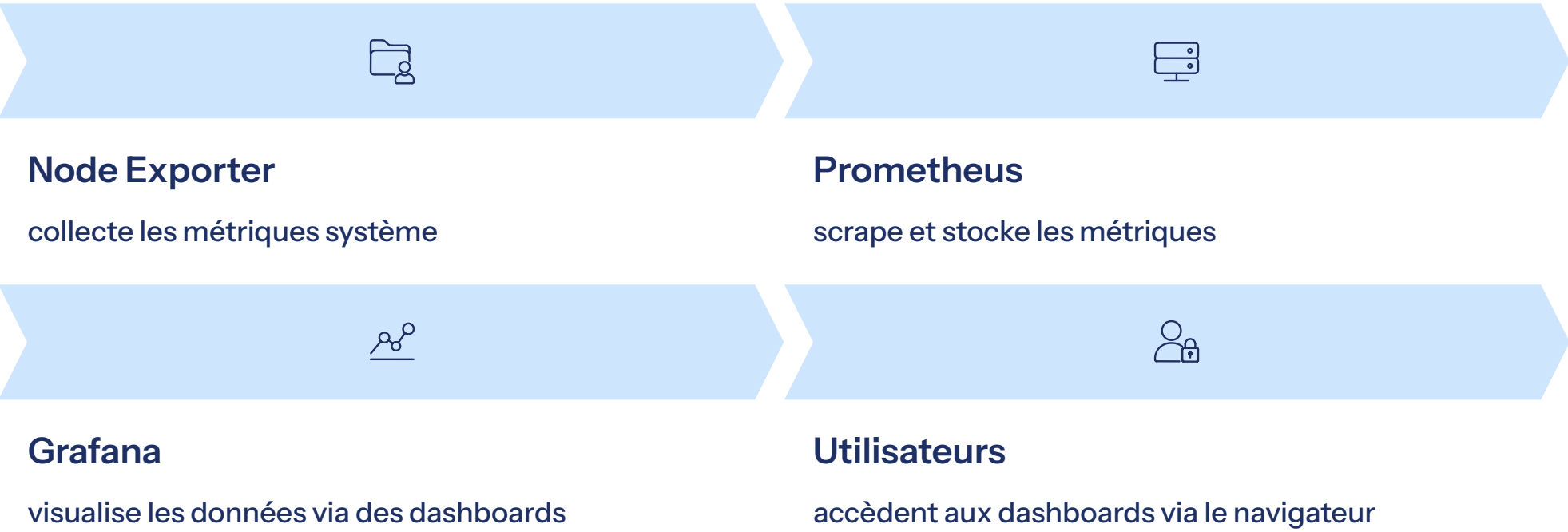
- Surveillance en temps réel des performances système
- Tableaux de bord personnalisés pour les métriques AWS EC2
- Automatisation complète du déploiement
- Architecture scalable et reproductible



# Architecture du Système



## Flux de données



# Structure du Projet

## Organisation des Fichiers

```
grafana-monitoring-ansible/
├── ansible.cfg
├── inventories/
│   ├── production/
│   │   ├── hosts
│   │   └── group_vars/
│   │       ├── all.yml
│   │       └── playbooks/
│   │           └── deploy-monitoring.yml
│   └── roles/
│       ├── grafana/
│       │   ├── tasks/
│       │   │   └── main.yml
│       │   ├── templates/
│       │   │   ├── aws-dashboard.json.j2
│       │   │   └── grafana.ini.j2
│       │   └── files/
│       │       └── datasource.yml
│       ├── prometheus/
│       │   ├── tasks/
│       │   │   └── main.yml
│       │   ├── templates/
│       │   │   └── prometheus.yml.j2
│       │   ├── handlers/
│       │   │   └── main.yml
│       │   └── node_exporter/
│       │       ├── tasks/
│       │       │   └── main.yml
│       │       ├── handlers/
│       │       │   └── main.yml
│       │       └── main.yml
│       └── main.yml
```

## Description des Fichiers

### Configuration Ansible

- **ansible.cfg** : Configuration globale d'Ansible
- **inventories/production/hosts** : Définition des serveurs cibles
- **group\_vars/all.yml** : Variables communes à tous les hôtes

### Playbooks

- **playbooks/deploy-monitoring.yml** : Playbook principal de déploiement

### Rôles

#### Grafana :

- **tasks/main.yml** : Installation et configuration
- **templates/grafana.ini.j2** : Configuration serveur
- **templates/aws-dashboard.json.j2** : Dashboard de monitoring AWS
- **files/datasource.yml** : Configuration source de données Prometheus

#### Prometheus :

- **tasks/main.yml** : Installation et configuration
- **templates/prometheus.yml.j2** : Configuration des jobs de scraping
- **handlers/main.yml** : Gestionnaires de services

#### Node Exporter :

- **tasks/main.yml** : Installation et configuration
- **handlers/main.yml** : Gestionnaires de services

# Prérequis et Installation

## Serveur Cible

- **Système d'exploitation** : Amazon Linux 2023
- **Utilisateur** : ec2-user avec privilèges sudo
- **Ports ouverts** : 3000 (Grafana), 9090 (Prometheus), 9100 (Node Exporter)
- **Espace disque** : Minimum 37GB libre
- **Mémoire** : Minimum 8GB RAM

## Machine de Déploiement

- **Ansible** : Version 2.9+
- **Python** : Version 3.6+
- **Accès SSH** : Clé SSH configurée

## Configuration SSH

```
# Générer une clé SSH (si nécessaire)
ssh-keygen -t rsa -b 4096 -f ~/.ssh/my_new_key

# Copier la clé publique vers le serveur
ssh-copy-id -i ~/.ssh/my_new_key.pub ec2-user@3.105.85.131
```

## Installation des Dépendances

```
# Sur la machine de déploiement
sudo yum install ansible python3-pip -y

# Vérifier l'installation
ansible --version
python3 --version
```

# Configuration Détaillée

1

## Configuration Ansible (ansible.cfg)

```
[defaults]
host_key_checking = False
inventory =
inventories/production/hosts
private_key_file =
~/.ssh/my_new_key
remote_user = ec2-user
roles_path = roles

[privilege_escalation]
become = True
become_method = sudo
```

2

## Inventaire (inventories/production/ hosts)

```
[monitoring_servers]
3.105.85.131

[monitoring_servers:vars]
ansible_python_interpreter=
/usr/bin/python3
```

3

## Variables Globales (group\_vars/all.yml)

```
# Configuration Grafana
grafana_admin_user:
"admin"
grafana_admin_password:
"admin"
grafana_version: "10.2.0"
grafana_port: 3000

# Configuration Prometheus
prometheus_version:
"2.47.0"
prometheus_port: 9090

# Configuration Node
Exporter
node_exporter_version:
"1.6.1"
node_exporter_port: 9100

# Configuration Domaine
domain_name:
"3.105.85.131"
```

# Déploiement

## Processus de Déploiement



## Étapes du Déploiement





# Accès aux Services

## URLs des Services

Service	URL	Port	Description
Grafana	<a href="http://3.105.85.131:3000">http://3.105.85.131:3000</a>	3000	Interface de visualisation
Prometheus	<a href="http://3.105.85.131:9090">http://3.105.85.131:9090</a>	9090	Interface de requêtes
Node Exporter	<a href="http://3.105.85.131:9100/metrics">http://3.105.85.131:9100/metrics</a>	9100	Métriques brutes



### Identifiants Grafana

- **Utilisateur** : admin
- **Mot de passe** : admin

**NB** : Changer le mot de passe après la première connexion si vous voulez !

## Dashboard AWS EC2

Le dashboard inclut les métriques suivantes :



### Utilisation CPU

Pourcentage d'utilisation du processeur



### Utilisation Mémoire

Pourcentage de mémoire utilisée



### I/O Disque

Taux de lecture/écriture en bytes/s



### Trafic Réseau

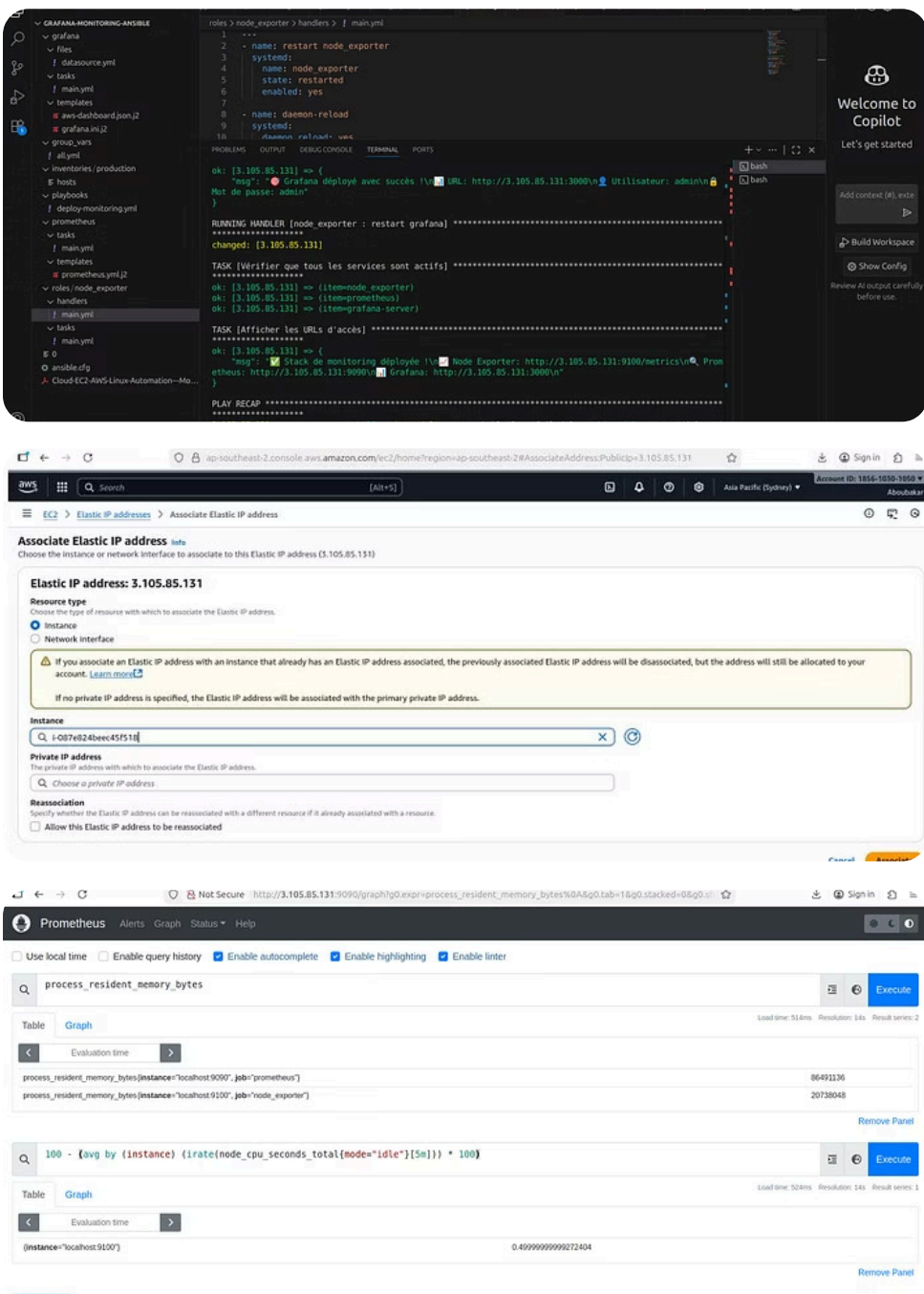
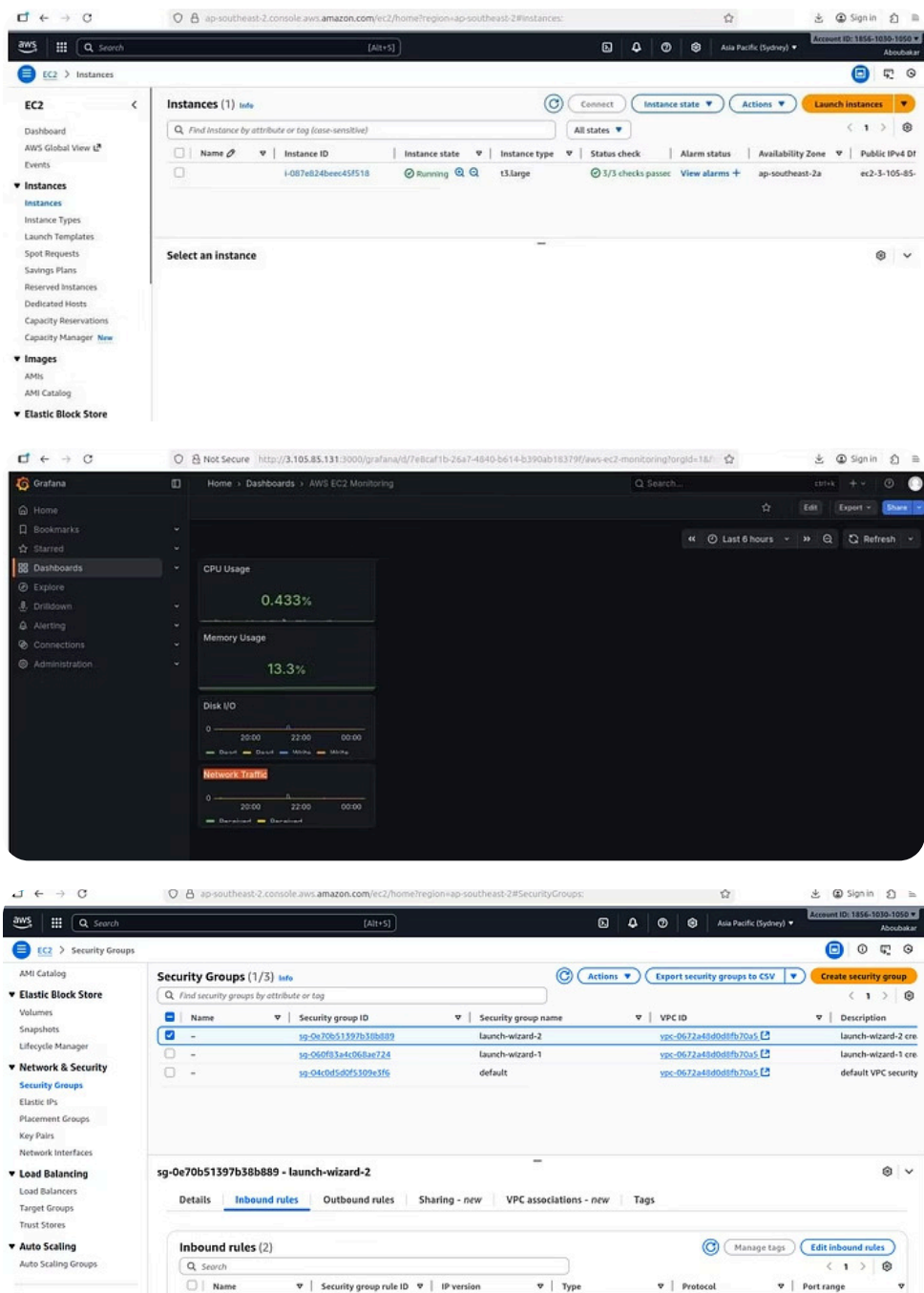
Données reçues/transmises en bytes/s

## Panels du Dashboard

1. **CPU Usage** : Métrique statistique avec unité en pourcentage
2. **Memory Usage** : Métrique statistique avec unité en pourcentage
3. **Disk I/O** : Série temporelle pour lecture/écriture
4. **Network Traffic** : Série temporelle pour réception/transmission

# Résultats et Conclusion

## Résultats du Déploiement



## Configuration AWS EC2

Pour rendre les liens de votre instance EC2 accessible depuis internet il faut créer un Elastic IP, l'associer à votre instance avec le bouton Actions cliquer sur Associate Elastic IP Address

Choisir le Security Groups en lien avec votre instance et permettre le trafic entrant pour rendre l'IP Elastic Publique et accéder aux applications et liens depuis internet

## Requêtes Prometheus

- 1- Mémoire résidentielle** : La première requête donne la quantité de mémoire résidentielle utilisée par un processus spécifique. La mémoire résidentielle est la mémoire utilisée par un programme (ou processus) et qui réside réellement en mémoire physique (RAM), par opposition à la mémoire virtuelle (qui peut être échangée avec le disque dur)
- 2 – Consommation mémoire** : La deuxième permet de suivre la consommation mémoire de Prometheus et Node Exporter en temps réel, afin de t'assurer que ces processus n'utilisent pas de manière excessive de la mémoire.

## Conclusion

Cette stack de monitoring fournit une solution complète pour surveiller les performances des instances AWS EC2. L'automatisation via Ansible assure un déploiement reproductible et fiable, tandis que la modularité des rôles permet une maintenance aisée.

3	3	100%
Services Déployés	Ports Configurés	Automatisation
Grafana, Prometheus, Node Exporter	3000, 9090, 9100	Déploiement complet via Ansible