

Система автоматизованого розгортання моніторингу із застосуванням експертних СИСТЕМ

Виконав:

студент II курсу

групи КА-83мп

Шевчук Назар Миколайович

Науковий керівник:

к.т.н., доцент Дідковська М. В.

Актуальність дослідження

- Стрімкий ріст компонентів інфраструктури потребує впровадження якісної системи моніторингу.
- Задля виявлення та запобігання збоїв компонентів ІТ-інфраструктури існує гостра необхідність оперативного виявленні недоліків інфраструктури та аномалій у даних.
- Необхідність автоматизованого розгортання.

- **Мета дослідження**

Розробити систему автоматизованого розгортання моніторингу із застосуванням експертних систем.

- **Об'єкт дослідження**

Засоби управління конфігурацій, система моніторингу побудована із застосуванням експертних систем.

- **Предмет дослідження**

Система збору метрик Prometheus Exporter, система збереження метрик TSDB Prometheus, система візуалізації Grafana, методи розробки експертних систем, засіб управління конфігурації Ansible.

Завдання

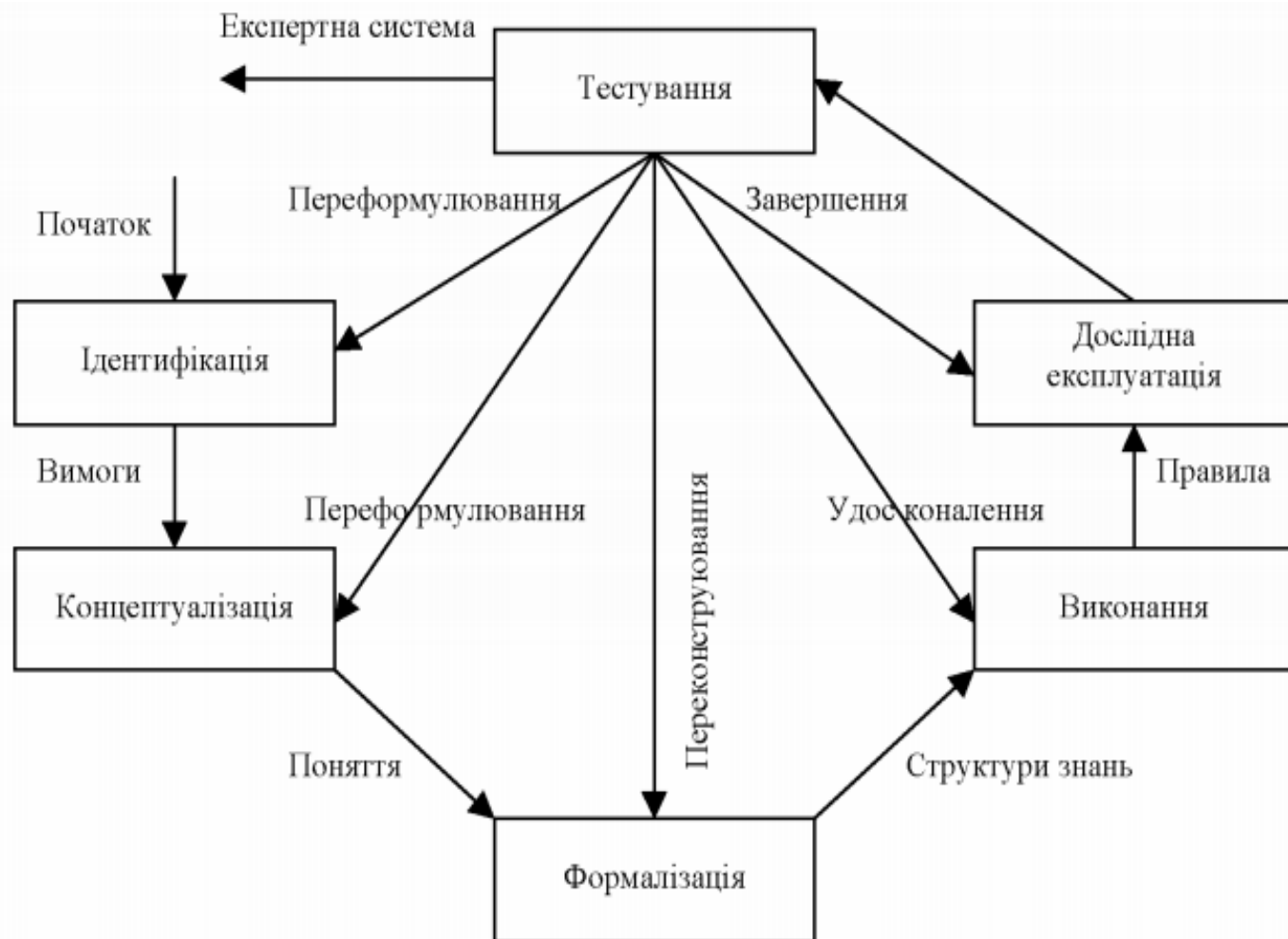
Дослідити особливості розробки та побудови експертних систем.

Дослідити особливості налаштування та імплементації систем моніторингу.

Дослідити засоби управління конфігурації.

Опираючись на проведені дослідження розробити сценарій автоматизованого розгортання експертної системи на основі інструментів моніторингу.

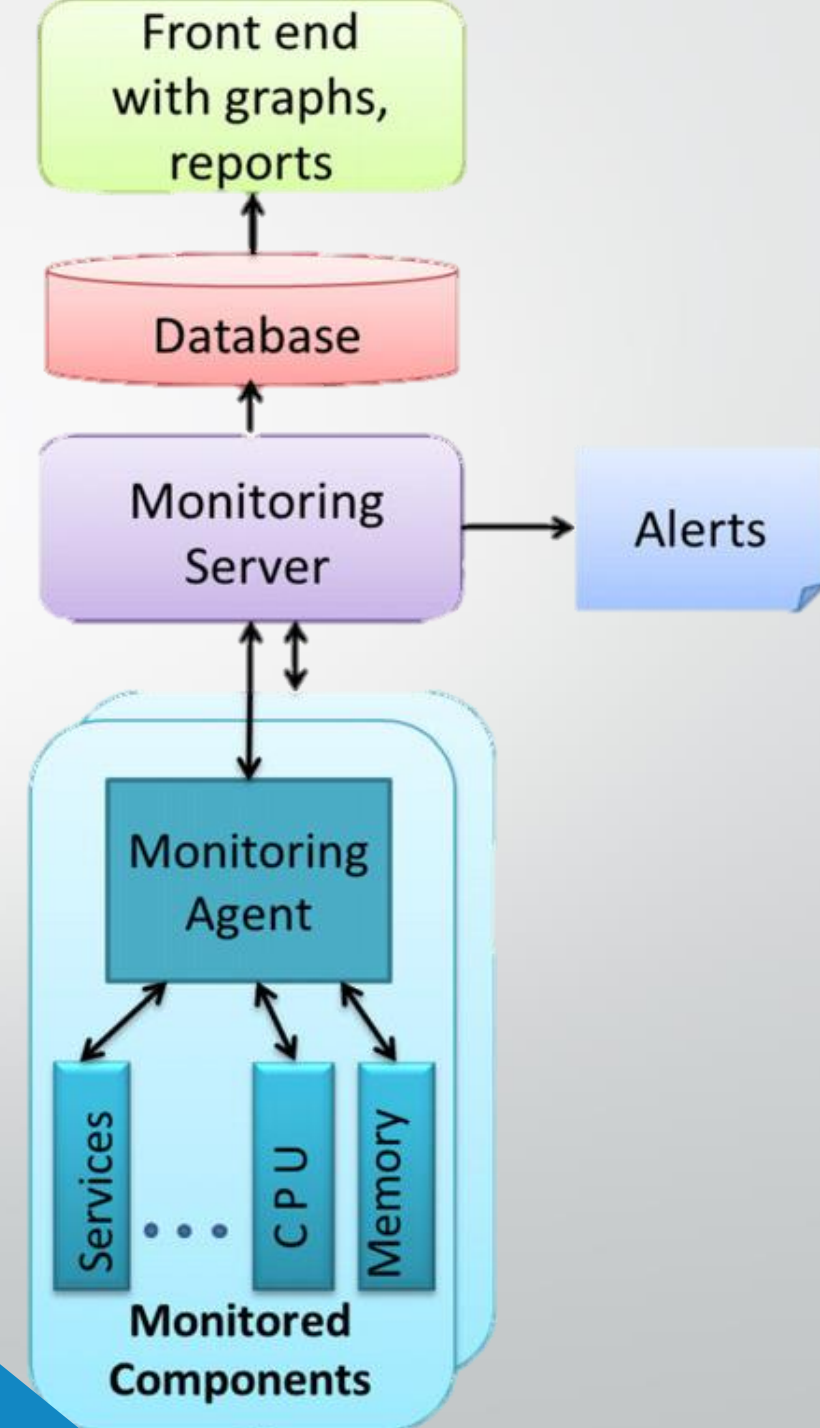
Етапи розробки експертної системи



Ідентифікація

Розроблена експертна система має виконувати наступні функції:

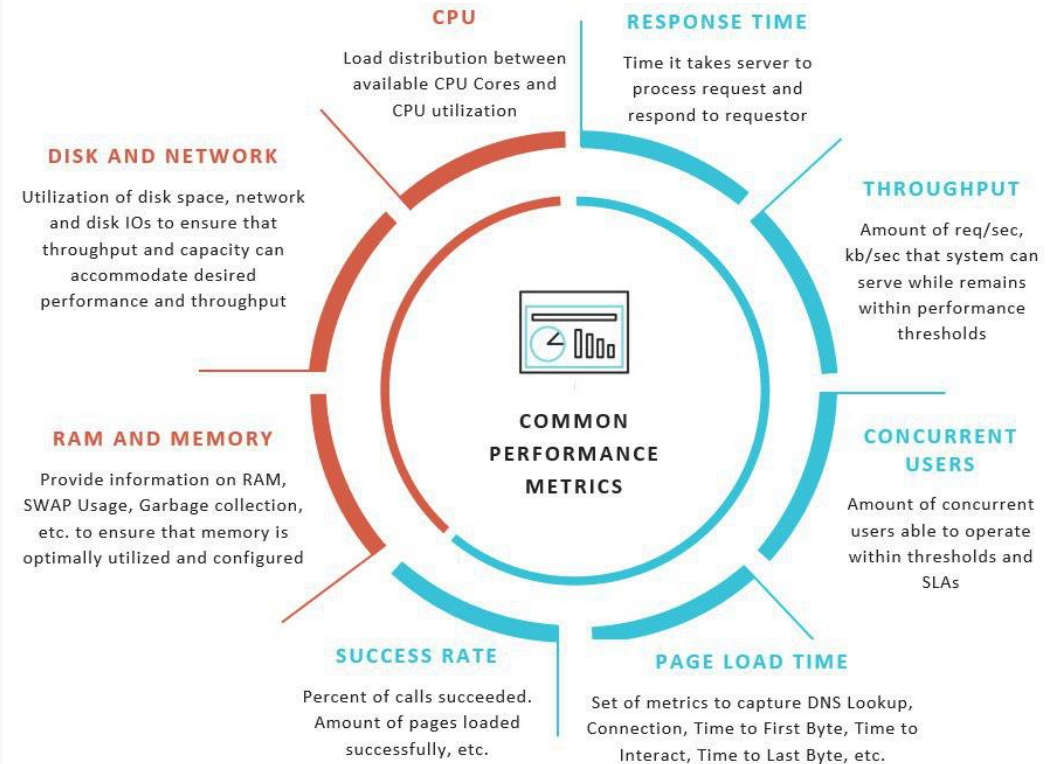
- Збір метрик з кінцевих вузлів
- Зберігання метрик
- Інтерпретація метрик налаштованими правилами
- Сповіщення користувача в разі виконання правил
- Візуалізація зібраних метрик засобами веб-інтерфейсу



Концептуалізація

Розроблена ЕС повинна мати змогу оперувати наступними показниками:

- Рівень навантаження на апаратне забезпечення серверів (ЦПУ, пам'ять, мережеві інтерфейси, дисковий простір)
- Показники кінцевих користувачів (швидкість завантаження, успішність завантаження, кількість запитів, кількість користувачів, швидкість відгуку системи)

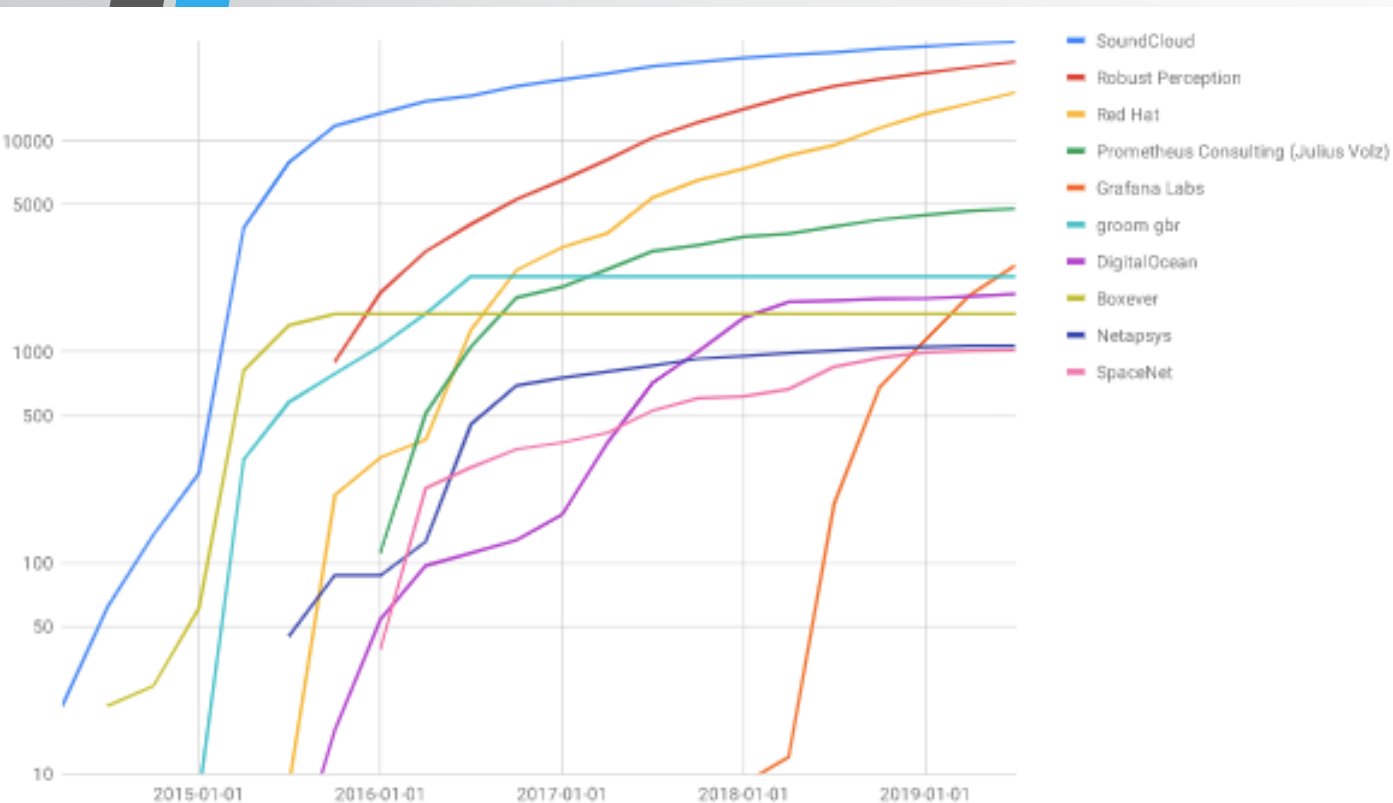


Формалізація. Вибір засобів розробки

	Nagios	Graphite	InfluxDB	OpenTSDB	Prometheus
Опис	Базується на свовіщеннях, скрипти перевірки	Базується на часових рядах, мові запитів та графіках	Часові ряди, непервні запити	Розподілені часові ряди, мова запитів, побудова графіків	Повний моніторинг, вбудоване стягування даних, мова запитів, графікування
Модель даних	Базується на хості, відсутня мова запитів	Зберігає чисельні зразки	Пари ключ-значення (таги)	Ключ-значення	Зберігає чисельні зразки
Зберігання даних	Немає сховища, плагіни зберігають дані	Часові ряди на локальному диску	Дерево структурованих логів із записом зверху	Інтегровано в Hadoop та HBase	Часові ряди на локальному диску
Переваги	Базовий моніторинг статичних систем	Кластерне рішення, довготривале зберігання даних	Логування подій, кластерне рішення	Довготривале зберігання даних в Hadoop	Багатша модель даних, підтримує мову запитів, легка інтеграція

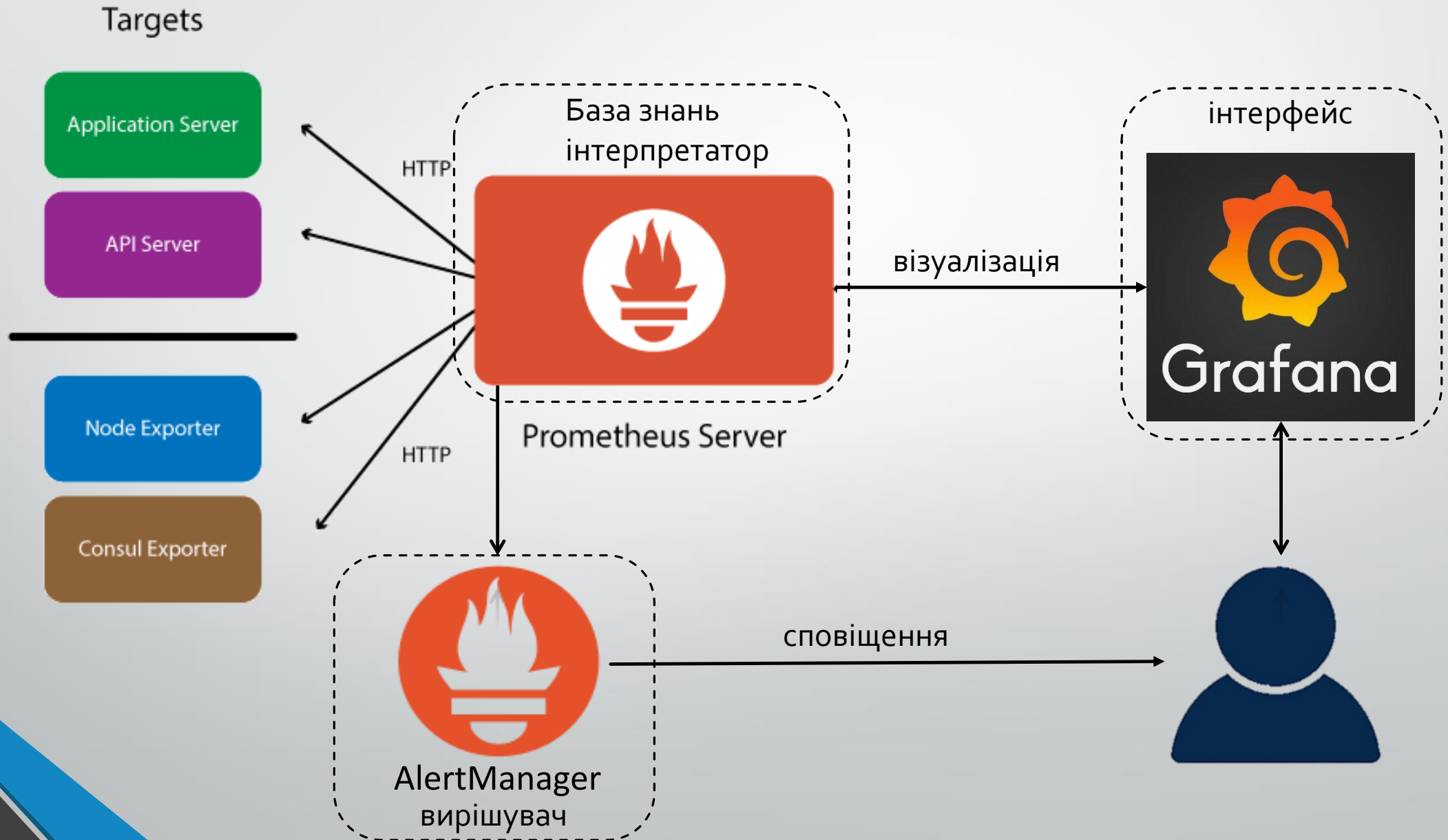
Тенденції використання Prometheus

6.3K contributors
13.5K code commits
7.2K pull requests
113K contributions
723 contributing companies

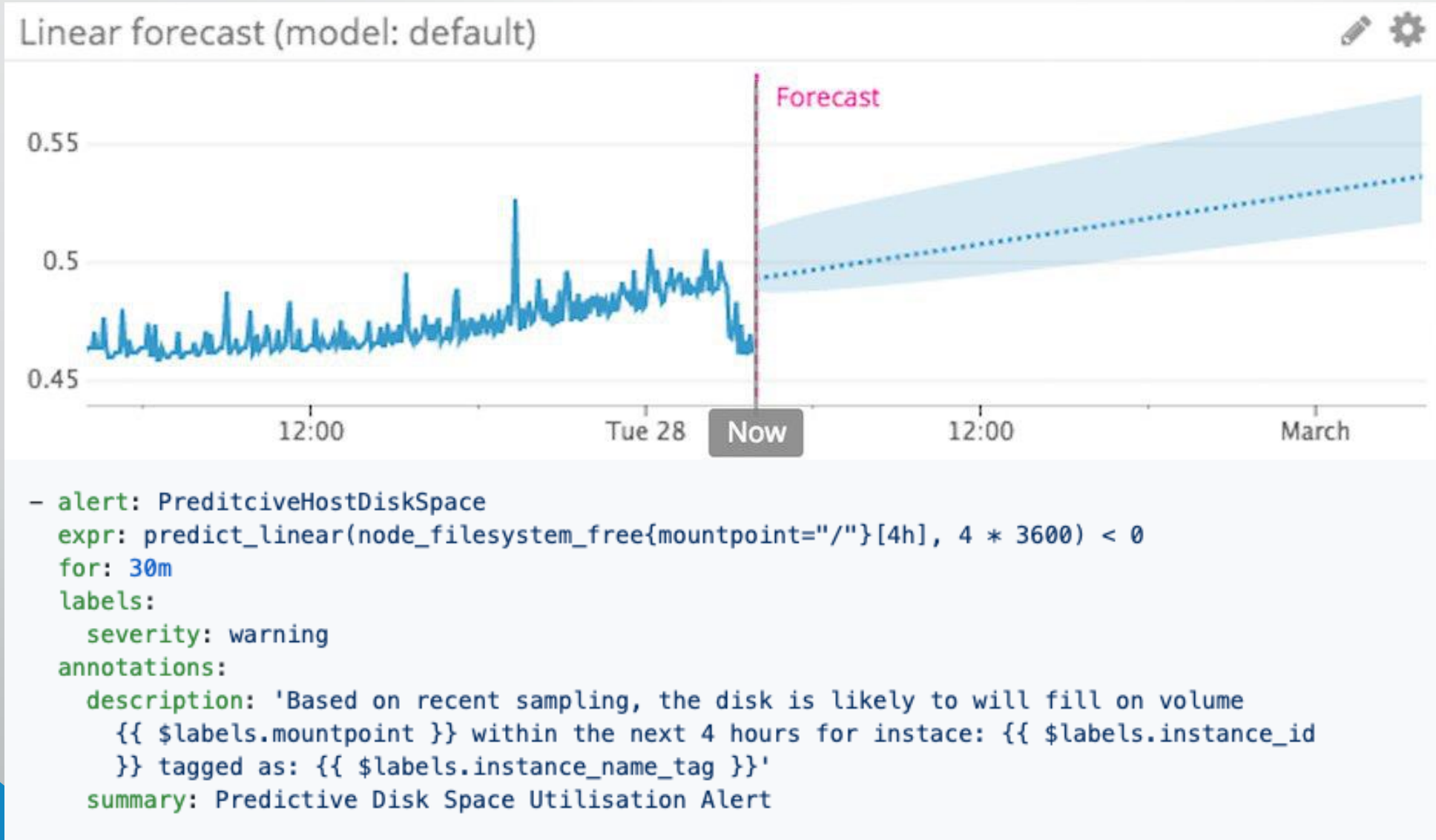


Ці статистичні дані були зібрані за допомогою інструменту DevStats, який CNCF (Cloud Native Computing Foundation) створив у співпраці з спільнотами проектів CNCF.

Виконання. Архітектура ES



Виконання. Прогнозування заповнення дискового простору





Виконання. Виявлення аномалій в кількості запитів до продуктового середовища в секунду

server down for more than 1 minute



Wellnessapi_Monitor APP 1:37 AM

[FIRING:1] (ExporterDown 10.133.21.126:9419 rabbitmq_exporter warning)

@channel Exporter down (instance 10.133.21.126:9419)

Prometheus exporter down

VALUE = 0

LABELS: map[__name__:up instance:10.133.21.126:9419 job:rabbitmq_exporter]

[FIRING:1] (host-down 10.133.21.126:9419 rabbitmq_exporter)

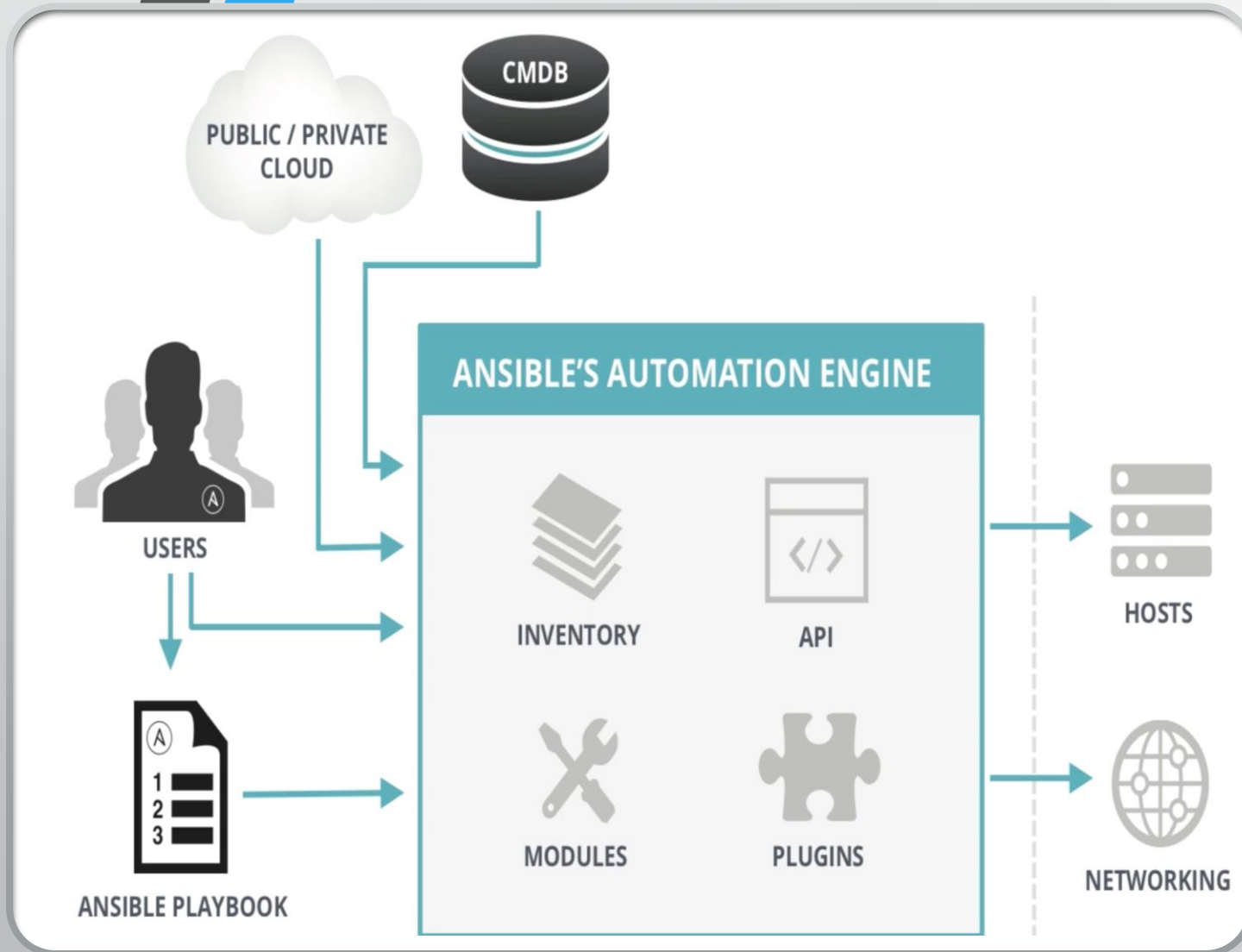
@channel

Server down for more than 1 minute

Тестування. Отримання сповіщення
в разі недоступності експортера

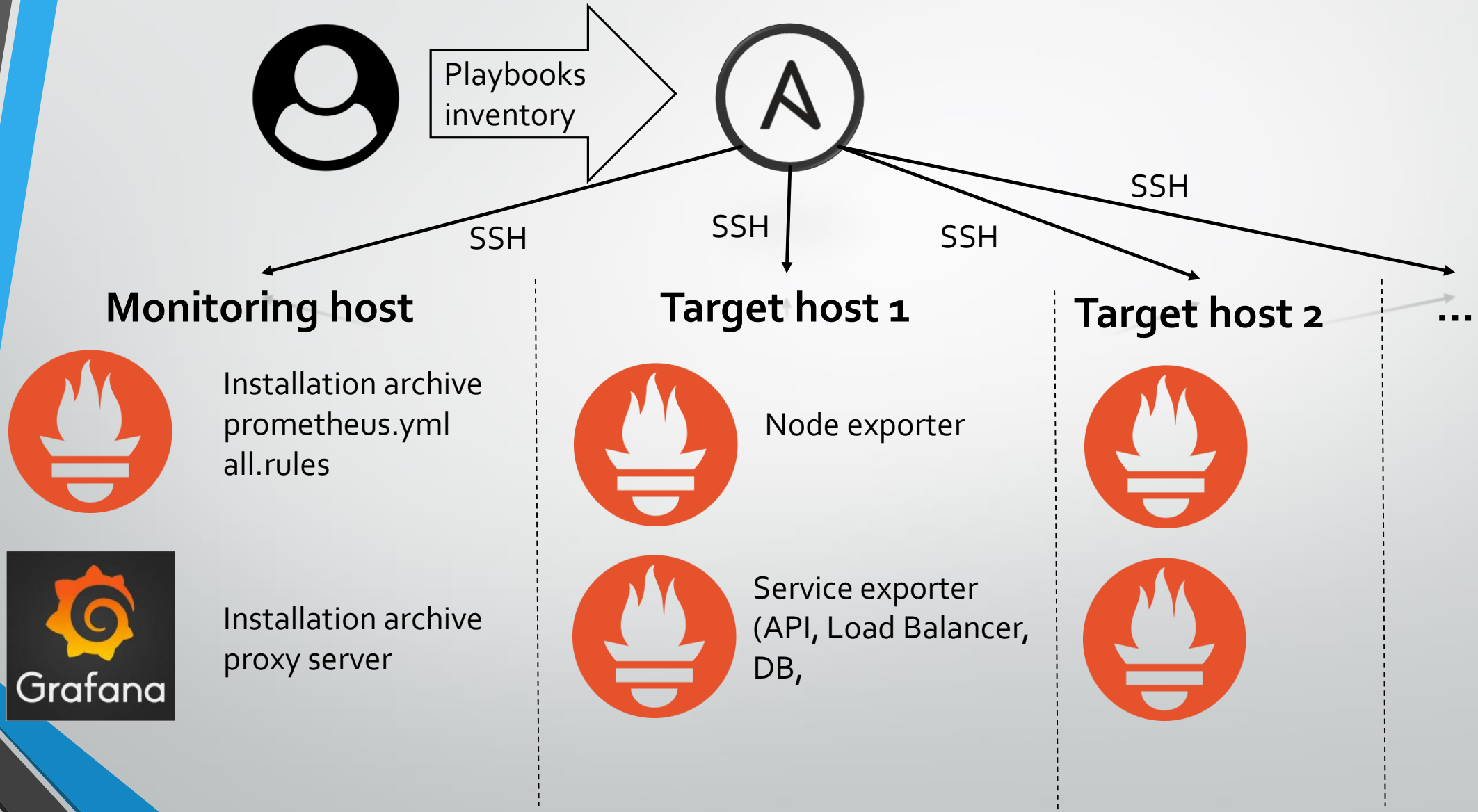
	ANSIBLE	PUPPET	CHEF
Мова написання скриптів (сценарію)	YAML	Спеціальний DSL на основі Ruby	Ruby
Необхідні компоненти інфраструктури	Сервер контролера застосовує конфігурацію на вузлах через SSH	Puppet-сервер синхронізує конфігурацію на Puppet-вузлах за допомогою Puppet-агентів	Робочі станції Chef підштовхують конфігурацію до Chef-серверу, звідки Chef-вузли буде оновлено
Потрібно спеціалізоване програмне забезпечення для вузлів	Ні (за винятком Python, який у більшості дистрибутивах ОС встановлений за замовчуванням).	Так.	Так.
Забезпечує централізований пункт контролю	Ні. Будь-який комп'ютер може бути контролером	Так, через Puppet-сервер	Так, через Chef-сервер
Термінологія сценаріїв	Playbook / Roles	Manifests / Modules	Recipes / Cookbooks
Виконання сценаріїв	Послідовне	Паралельне	Послідовне

Порівняльний аналіз засобів управління конфігурації



Архітектура Ansible

Принципи застосування



Подальші шляхи розвитку роботи

- Забезпечення безпеки експортера (автоматизоване налаштування фаєрволу)
- Налаштування додаткової аутентифікації до інтерфейсу.

Наукова новизна

- Запропоновано експертну систему для побудови серверного моніторингу.
- Запропоновано алгоритм управління конфігурацією для автоматизації розгортання системи в різних середовищах.

Практична цінність

- Запропоновано архітектуру системи.
- Розроблена система моніторингу яка здатна виявляти аномалії в продуктових даних.

Висновки

- Було досліджено застосування експертних систем при побудові серверного моніторингу.
- В результаті автоматизованого розгортання отримано працездатну кросс-платформенну систему моніторингу з якісною візуалізацією в Grafana.
- На прикладі побудови моніторингу досліджено можливість створення системи виявлення аномалій як результат застосування експертних систем.