



Introducción

¿Qué es
Prometheus?

¿Cómo se monitoriza
un nuevo target?

Motivaciones y Objetivos

Motivaciones:
Problemas con la
monitorización actual

Objetivos: ¿Qué se
pretende conseguir
con la app?

Proceso de desarrollo

Ansible
API en Python
Interfaz

Resultados y Conclusiones

Resultados:
Funcionamiento de la
aplicación
Trabajo futuro
Conclusiones: App vs
tradicional

GRAFANA, ALERT-MANAGER AND PROMETHEUS MANAGER

Vidal Peña, Arturo ¹ Pérez Costoya, Fernando ²

¹Escuela Técnica Superior de Ingeniería Informática
Universidad Politécnica de Madrid

²Departamento de Arquitectura y Tecnología de Sistemas Informáticos
Escuela Técnica Superior de Ingeniería Informática
Universidad Politécnica de Madrid

En colaboración con Accenture Technologies S.L.

Junio 2022



Introducción

¿Qué es Prometheus?

¿Cómo se monitoriza un nuevo target?

Motivaciones y Objetivos

Motivaciones:
Problemas con la monitorización actual

Objetivos: ¿Qué se pretende conseguir con la app?

Proceso de desarrollo

Ansible
API en Python
Interfaz

Resultados y Conclusiones

Resultados:
Funcionamiento de la aplicación

Trabajo futuro
Conclusiones: App vs tradicional

1 Introducción

- ¿Qué es Prometheus?
- ¿Cómo se monitoriza un nuevo target?

2 Motivaciones y Objetivos

- Motivaciones: Problemas con la monitorización actual
- Objetivos: ¿Qué se pretende conseguir con la app?

3 Proceso de desarrollo

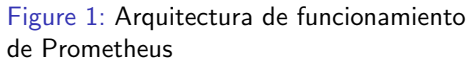
- Ansible
- API en Python
- Interfaz

4 Resultados y Conclusiones

- Resultados: Funcionamiento de la aplicación
- Trabajo futuro
- Conclusiones: App vs tradicional



- Resultados:
- Funcionamiento de la aplicación
- Trabajo futuro
- Conclusiones: App vs tradicional



- Servidor de prometheus
- Targets con etiquetas, agrupados por proyectos
- Reglas de alertado



¿Cómo se monitoriza un nuevo target?

Introducción

¿Qué es

Prometheus?

¿Cómo se monitoriza
un nuevo target?

Motivaciones y Objetivos

Motivaciones:

Problemas con la
monitorización actual

Objetivos: ¿Qué se
pretende conseguir
con la app?

Proceso de desarrollo

Ansible

API en Python

Interfaz

Resultados y Conclusiones

Resultados:

Funcionamiento de la
aplicación

Trabajo futuro

Conclusiones: App vs
tradicional

Forma estática:

```
- job: node exporter
static_configs:
- targets:
  - "localhost:9091"
  labels:
    jobname: node
    os: linux
```

Forma dinámica:

```
- job: node exporter
  file_sd_configs:
  - files:
    - "targets/*.json"
    refresh_interval: 2m
```

```
{
  "targets": [
    "localhost:9091"
  ],
  "labels": {
    "jobname": "node",
    "os": "linux"
  }
}
```



Motivaciones: Problemas con la monitorización actual

Introducción

¿Qué es Prometheus?

¿Cómo se monitoriza un nuevo target?

Motivaciones y Objetivos

Motivaciones:
Problemas con la monitorización actual

Objetivos: ¿Qué se pretende conseguir con la app?

Proceso de desarrollo

Ansible
API en Python
Interfaz

Resultados y Conclusiones

Resultados:
Funcionamiento de la aplicación
Trabajo futuro
Conclusiones: App vs tradicional

- 1 Proceso lento y repetitivo
- 2 Con etiquetas diferentes, hay que separar los targets
- 3 No hay automatizaciones oficiales
- 4 Alta probabilidad de introducir errores humanos
- 5 Con ficheros grandes, los errores son difíciles de detectar



Objetivos: ¿Qué se pretende conseguir con la app?

Introducción

¿Qué es Prometheus?

¿Cómo se monitoriza un nuevo target?

Motivaciones y Objetivos

Motivaciones:
Problemas con la monitorización actual

Objetivos: ¿Qué se pretende conseguir con la app?

- Automatizar el proceso de añadir targets
- Aumentar la rapidez y eficacia del proceso
- Disminuir el número de errores introducidos

Proceso de desarrollo

Ansible
API en Python
Interfaz

Resultados y Conclusiones

Resultados:
Funcionamiento de la aplicación
Trabajo futuro
Conclusiones: App vs tradicional



¿Cómo funciona Ansible en este proyecto?

Introducción

¿Qué es Prometheus?

¿Cómo se monitoriza un nuevo target?

Motivaciones y Objetivos

Motivaciones: Problemas con la monitorización actual

Objetivos: ¿Qué se pretende conseguir con la app?

Proceso de desarrollo

Ansible

API en Python

Interfaz

Resultados y Conclusiones

Resultados: Funcionamiento de la aplicación

Trabajo futuro

Conclusiones: App vs tradicional

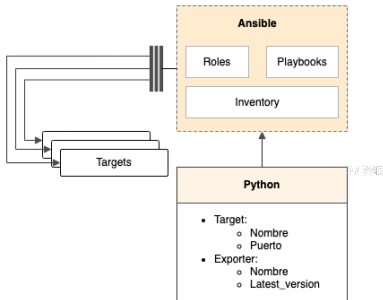


Figure 2: Diagrama ansible-node_exporter

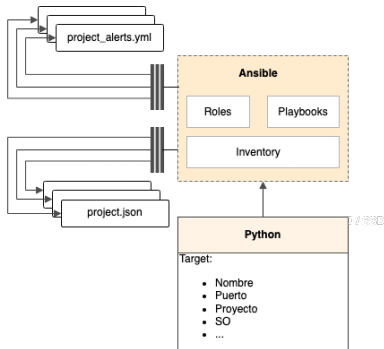


Figure 3: Diagrama ansible-ficheros_prometheus



API en Python

Introducción

¿Qué es Prometheus?

¿Cómo se monitoriza un nuevo target?

Motivaciones y Objetivos

Motivaciones:
Problemas con la monitorización actual

Objetivos: ¿Qué se pretende conseguir con la app?

Proceso de desarrollo

Ansible
API en Python
Interfaz

Resultados y Conclusiones

Resultados:
Funcionamiento de la aplicación

Trabajo futuro

Conclusiones: App vs tradicional

- Establece el backend
- Genera endpoints para conectar con frontend
- Ejecuta ansible para instalar node exporter en los targets
- Ejecuta ansible para añadir target a Prometheus
- Crea el exporter de versiones en el target
- Añade un fichero de targets y de reglas de alertado para cada proyecto



Interfaz

Introducción

¿Qué es
Prometheus?

¿Cómo se monitoriza
un nuevo target?

Motivaciones y Objetivos

Motivaciones:
Problemas con la
monitorización actual

Objetivos: ¿Qué se
pretende conseguir
con la app?

Proceso de desarrollo

Ansible
API en Python
Interfaz

Resultados y Conclusiones

Resultados:
Funcionamiento de la
aplicación
Trabajo futuro
Conclusiones: App vs
tradicional

- Hecha con ElectronJS
- Simple y práctica
- Permite:
 - 1 Crear proyectos
 - 2 Añadir targets
 - 3 Buscar targets por proyecto

Resultados: Funcionamiento de la aplicación

Mejor poner fotos, el vídeo puede salirse de tiempo fácilmente.



Trabajo futuro

Introducción

¿Qué es Prometheus?

¿Cómo se monitoriza un nuevo target?

Motivaciones y Objetivos

Motivaciones:
Problemas con la monitorización actual

Objetivos: ¿Qué se pretende conseguir con la app?

Proceso de desarrollo

Ansible
API en Python
Interfaz

Resultados y Conclusiones

Resultados:
Funcionamiento de la aplicación

Trabajo futuro

Conclusiones: App vs tradicional

- Trabajar con más exporters
- Añadir más servidores de Prometheus, Alertmanager y Grafana
- Monitorización en la nube
- Creación de dashboards en Grafana mediante la app



Conclusiones: App vs tradicional

Introducción

¿Qué es Prometheus?
¿Cómo se monitoriza un nuevo target?

Motivaciones y Objetivos

Motivaciones:
Problemas con la monitorización actual
Objetivos: ¿Qué se pretende conseguir con la app?

Proceso de desarrollo

Ansible
API en Python
Interfaz

Resultados y Conclusiones

Resultados:
Funcionamiento de la aplicación
Trabajo futuro
Conclusiones: App vs tradicional

Objetivos cumplidos:

- ✓ La opción de introducir errores se ve reducida a la insertada por el usuario
- ✓ Los targets se añaden automáticamente
- ✓ La velocidad en el proceso aumenta considerablemente