**Stack Actual de PlatSentinel**

**Resumen de componentes principales:**

* **Toquen de Antesala:** Microservicio en Docker, Python 3.11-alpine, usa ClamAV, PyMuPDF, python-magic. Comunicación vía RabbitMQ (entrada) y API REST (salida) con JWT.
* **Integración WhatsApp:** Python (FastAPI), integración Twilio API para recibir comandos y activar análisis mediante mensajería.
* **Dashboard Web:** Visualización de servicios con D3.js y Sigma.js, autenticación OIDC (Ory Hydra), editor de reglas con Monaco Editor.
* **Infraestructura como código:** Terraform para AWS EKS y VPC, despliegue automatizado con GitHub Actions, backups automatizados.
* **Pipeline CI/CD:** GitHub Actions para build y despliegue de contenedores, reinicio de despliegues en Kubernetes.

**Análisis del Stack**

**Puntos Fuertes**

* **Arquitectura modular:** Uso de microservicios en Docker facilita escalabilidad y mantenimiento.
* **Automatización:** CI/CD y Terraform permiten despliegues reproducibles y seguros.
* **Integración de mensajería:** WhatsApp como vía de interacción ágil para usuarios y técnicos.
* **Visualización avanzada:** Uso de D3.js y Sigma.js para mapear servicios gestionados.
* **Seguridad:** Uso de JWT, ClamAV para análisis, y Vault para gestión de secretos.

**Áreas Mejorables y Recomendaciones**

**1. Seguridad de Contenedores Docker**

* Ejecutar contenedores siempre como usuario no root y con privilegios mínimos[1](https://talent500.com/blog/modern-docker-best-practices-2025/)[2](https://cloudnativenow.com/topics/cloudnativedevelopment/docker/docker-security-in-2025-best-practices-to-protect-your-containers-from-cyberthreats/)[3](https://www.sentinelone.com/cybersecurity-101/cloud-security/container-security-best-practices/).
* Usar imágenes base firmadas y escaneadas regularmente con herramientas como Trivy o Docker Scout.
* Habilitar archivos de sistema en modo solo lectura y limitar recursos (CPU/memoria).
* Implementar auditoría continua (CIS Docker Benchmark) y monitorización con Falco o Sysdig.

**2. Orquestación y Observabilidad**

* Incorporar Prometheus y Grafana para métricas y alertas de todos los microservicios.
* Añadir herramientas de tracing distribuido (por ejemplo, Jaeger) para depurar flujos complejos.
* Centralizar logs con ELK (Elasticsearch, Logstash, Kibana) o Loki.

**3. API y Backend**

* FastAPI es una excelente elección para APIs modernas, por rendimiento y documentación automática[4](https://betterstack.com/community/guides/scaling-python/flask-vs-fastapi/)[5](https://wappnet.com/blog/fastapi-vs-flask-which-framework-should-you-use-in-2025/).
* Añadir validación estricta de entradas y salidas, y pruebas automáticas integradas en el pipeline.
* Revisar la integración de Ory Hydra para asegurar cumplimiento de mejores prácticas de OAuth2/OIDC.

**4. Seguridad y Zero Trust**

* Aplicar el principio Zero Trust: segmentar la red, restringir accesos y auditar permisos de cada microservicio[6](https://www.cwstechnology.com/blog/cybersecurity-best-practices-custom-software-2025/).
* Usar mTLS para la comunicación entre servicios internos.
* Firmar y verificar imágenes de Docker antes del despliegue (Docker Content Trust).

**5. Experiencia de Usuario y Extensibilidad**

* Permitir notificaciones multi-canal (además de WhatsApp, correo, Telegram, app móvil).
* Habilitar la integración sencilla de nuevos tokens y servicios externos mediante un sistema de plugins.
* Añadir soporte para reportes exportables (PDF, CSV) y APIs para integración con SIEM o plataformas de terceros.

**6. Gestión de Infraestructura**

* Revisar el uso de autoescalado en EKS/GKE para optimizar costes y disponibilidad.
* Automatizar pruebas de restauración de backups y políticas de retención de datos.

**Tabla de Mejoras Propuestas**

| **Área** | **Mejora Recomendada** |
| --- | --- |
| Seguridad Docker | Usuario no root, imágenes firmadas, escaneo continuo, auditoría CIS Docker, archivos solo lectura |
| Observabilidad | Prometheus, Grafana, tracing distribuido, centralización de logs |
| API/Backend | Validación estricta, pruebas automáticas, mejores prácticas OAuth2/OIDC |
| Zero Trust | Segmentación, mTLS, permisos mínimos, firma de imágenes |
| Experiencia de usuario | Notificaciones multi-canal, plugins para tokens, reportes exportables, integración SIEM |
| Infraestructura | Autoescalado, automatización de pruebas de backup, optimización de recursos |

**Conclusión**

El stack de PlatSentinel es moderno, modular y alineado con las tendencias de ciberseguridad y DevSecOps para 2025. Aplicando las mejoras propuestas, el proyecto ganará en robustez, seguridad, escalabilidad y facilidad de integración, posicionándose como una solución avanzada y competitiva para la gestión centralizada de seguridad en empresas pequeñas y servicios técnicos[7](https://heimdalsecurity.com/blog/best-cybersecurity-platforms/)[8](https://www.sentinelone.com/cybersecurity-101/cybersecurity/cyber-security-software/)[1](https://talent500.com/blog/modern-docker-best-practices-2025/)[2](https://cloudnativenow.com/topics/cloudnativedevelopment/docker/docker-security-in-2025-best-practices-to-protect-your-containers-from-cyberthreats/)[3](https://www.sentinelone.com/cybersecurity-101/cloud-security/container-security-best-practices/).

1. <https://talent500.com/blog/modern-docker-best-practices-2025/>
2. <https://cloudnativenow.com/topics/cloudnativedevelopment/docker/docker-security-in-2025-best-practices-to-protect-your-containers-from-cyberthreats/>
3. <https://www.sentinelone.com/cybersecurity-101/cloud-security/container-security-best-practices/>
4. <https://betterstack.com/community/guides/scaling-python/flask-vs-fastapi/>
5. <https://wappnet.com/blog/fastapi-vs-flask-which-framework-should-you-use-in-2025/>
6. <https://www.cwstechnology.com/blog/cybersecurity-best-practices-custom-software-2025/>
7. <https://heimdalsecurity.com/blog/best-cybersecurity-platforms/>
8. <https://www.sentinelone.com/cybersecurity-101/cybersecurity/cyber-security-software/>
9. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/49893040/54ccb211-f8c7-491f-a743-8c15e22dd2c8/Stack_1.docx>
10. <https://nihartechnologies.org/how-can-i-manage-it-services-in-business-factor/>
11. <https://dev.to/abubakersiddique761/2025s-must-know-tech-stacks-4b74>
12. <https://www.readability.com/top-7-programs-in-software-cyber-security-cloud-in-2025>
13. <https://sprinto.com/blog/cybersecurity-stack/>
14. <https://thectoclub.com/tools/best-cybersecurity-software/>
15. <https://www.contentful.com/blog/fastapi-vs-flask/>
16. <https://cheatsheetseries.owasp.org/cheatsheets/Docker_Security_Cheat_Sheet.html>
17. <https://dev.to/leapcell/is-flask-dead-is-fastapi-the-future-5glc>
18. <https://www.nucamp.co/blog/coding-bootcamp-cybersecurity-2025-what-are-the-projected-major-cybersecurity-trends-for-2025>
19. <https://www.suse.com/c/container-security-best-practices/>
20. <https://www.reddit.com/r/flask/comments/1jht6e5/is_flask_still_a_good_choice_in_2025/>
21. <https://thinksys.com/devops/docker-best-practices/>