Lectura 5:

Make para devops y devsecops

1. Que es make y como se lee un Makefile

- Make orquesta objetos, dependencias para construir proyectos reproducibles.
- Con marcas de tiempo make decide que esta desactualizado, si un requisito cambió.
- Las reglas de patron y variables realizan tareas.
- help no es un adorno, transforma el makefile en una interfaz de usuario técnica.
- Makefile se autodocumenta, muestra que comandos existen, para que sirven y como usarlos.

2. Make en DevOps

- busca automatizar y reducir ambigüedad y fricción
- El makefile puede cubrir 3 momentos: preparar, verificar, empaquetar/operar

a. Preparar: equivalente a build
- Al invocar make prepare solo se ejecutan los pasos de instalación.
- Los pasos deben ser idempotentes
- Si se cambia la version de python o deseas regerar el venv, puedes forzar la reconstrucción.

b. Verificar: smoke test pragmáticos
- Probar rápido que algo responde antes de seguir.
- PORT, MESSAGE, RELESE son variables.

C. Empaquetar/Operar: proxy reverse + servicio del sistema
- El pack no crea un binario, empaqueta.

- TLS cifra la comunicacion entre el usuario y el servidor.
- Con systemd la aplicación deja de depender de la terminal y se ejecuta como servicio del sistema
- Este enfoque estandariza despliegue y operación, facilita auditoría con journalctl

3. DevSecOps: Seguridad desde el diseño

- Aquí no solo importa que un servicio responda, nos interesa como responde.
- Debe aplicarse TLS, mínimo privilegio en seguridad
- Ser observable, logs/metricas/traces
- También debe tener resiliencia, escalabilidad, reproducibilidad, mantenible y con costos controlados.
- Todo esto garantiza confiabilidad, diagnostico rapido y despliegues producibles

4. Variables, flexibilidad y multiplataforma

- Las variables llevan el principio 12-factor al makefile
- Se puede cambiar las variables sin modificar código

5. Targets abstractos, .Phony

- .Phony evita colisiones con archivos homónimos y asegura ejecución.
- Cada target se auto_explica se descubre con make help.

6. Caché incremental y control del ciclo de vida

- Para desmontar lo desplegado existe la fase de limpieza, la cual deja el sistema listo para repetir.
- Esto detiene y deshabilita el servicio, borra la unidad y conserva certificados lo que seria un rollback limpio
- rollback es volver a un estado seguro sin dejar residuos operativos

7. DNS y nombres locales

- Este target automatiza el mapeo local del nombre al loopback
- Resuelve miapp.local a 127.0.0.1

8. .phony y ayuda como contrato de uso

- Gracias al objetivo help, que recorre y extrae las descripciones marcadas con ## usando grep
- El archivo se vuelve documentación Ejecutable ya que muestra los comandos, Para que sirven y como usarlos sin abrir archivos.