Proyecto Final: Caso de Estudio DevSecOps en el Pipeline

1. El Caso de Estudio: "FastTicket"

Usted y su equipo han sido contratados como consultores DevSecOps para modernizar el ciclo de desarrollo de una startup llamada **FastTicket**. FastTicket es una plataforma de venta de boletos en línea (e-commerce) que maneja eventos en vivo (conciertos, deportes).

Requisitos del Negocio y Técnicos

1. **Velocidad es Crítica (CI/CD):** Los eventos se anuncian con poca antelación, por lo que las correcciones de errores y las nuevas características (como nuevos métodos de pago o integración con billeteras digitales) deben desplegarse a producción **en menos de 30 minutos** una vez que el código es aprobado.
2. **Máxima Seguridad:** Al manejar datos de tarjetas de crédito y tokens de acceso (PCI-DSS), la aplicación tiene requisitos de seguridad extremadamente altos. **No puede haber vulnerabilidades críticas o de alta severidad** en el código o en el entorno.
3. **Arquitectura:**
    - La aplicación es una API REST (Node.js/Python) y un *frontend* (React).
    - Todo está alojado en contenedores Docker y orquestado con Kubernetes.
    - La base de datos es PostgreSQL.
4. **Entorno de Nube:** La empresa utiliza **AWS** como su proveedor de nube principal.

El Problema

Actualmente, FastTicket usa un proceso manual: el código se pasa de Dev a Staging sin pruebas automáticas de seguridad y el despliegue a Producción lo hace manualmente un solo ingeniero. Este proceso es lento y ha provocado *bugs* graves y, más recientemente, una alerta de seguridad por una **librería desactualizada con un CVE conocido**.

Su misión es diseñar una arquitectura de *pipeline* CI/CD automatizada y segura (DevSecOps) que cumpla con los requisitos de velocidad y seguridad.

2. Requisitos del Proyecto y Decisiones Estratégicas

El equipo debe documentar y justificar las decisiones tomadas en las siguientes áreas, basándose en la teoría del Módulo 1 (Shift-Left, Automatización, Hardening, SAST vs DAST).

A. Diseño de Entornos

Justifique la configuración de sus entornos.

1. **Entorno de Desarrollo (Dev):** ¿Qué tipo de pruebas (unitarias, locales) se ejecutan aquí? ¿Cómo se mantiene el *Hardening* de este entorno (e.g., uso de credenciales de prueba, recursos limitados)?
2. **Entorno de Staging/QA:** ¿Cómo se habilita este entorno? ¿Es un clon exacto de Producción? ¿Qué tipo de pruebas de seguridad **DAST** se ejecutan aquí?
3. **Entorno de Producción (Prod):** ¿Qué estrategias de *Hardening* final se aplican justo antes del *deploy*? ¿Cómo se garantiza la seguridad de la red y el acceso a los secretos?

B. Selección y Justificación de Herramientas

Proponga una herramienta para cada función (no necesita usar nombres reales, pero sí describir su tipo y justificar la elección):

1. **CI/CD Orchestration:** (E.g., Jenkins, GitLab CI, GitHub Actions, AWS CodePipeline).
2. **SAST:** ¿Por qué su herramienta SAST es adecuada para el lenguaje de FastTicket (Node.js/Python)?
3. **DAST:** ¿Qué herramienta DAST usarán y por qué es vital usarla en el entorno de Staging/QA?
4. **Análisis de CVE's (SCA):** ¿Dónde se integra esta herramienta en el *pipeline* y por qué es su **primera línea de defensa** contra dependencias vulnerables?
5. **Hardening / Seguridad de Contenedores:** ¿Qué herramienta usan para escanear y asegurar las imágenes de Docker antes de que lleguen a Producción?

C. Construcción del Pipeline (Stages y Gates)

Describa y ordene cronológicamente las etapas del *pipeline* de la rama principal (`main` o `master`).

#	Stage (Etapa)	Tareas Críticas (Acciones a realizar)	Gate (Criterio de Aprobación)

1	**CODE (Shift-Left)**	Ejecución de Pruebas Unitarias.	Cobertura de pruebas 80%
2	**BUILD & SCA**	Compilación de la imagen Docker. Ejecución de Análisis de CVE's (SCA).	Cero CVE's **críticos** y **altos** encontrados en las dependencias.
3	**SAST**	Escaneo Estático del código fuente.	Cero vulnerabilidades de alta severidad encontradas por SAST.
4	**DEPLOY to STAGING**	Despliegue en el entorno de Staging/QA. Habilitar entorno de pruebas.	Despliegue exitoso, servicio levantado.
5	**TEST & DAST**	Ejecución de Pruebas Funcionales (automatizadas). Ejecución de Análisis Dinámico (DAST).	Cero fallos en pruebas funcionales. Cero vulnerabilidades de alta o media severidad de DAST.
6	**HARDENING Final**	Aplicación de la configuración de *Hardening* del servidor y red.	Chequeo automático de políticas de seguridad.
7	**DEPLOY to PROD**	Despliegue final en el entorno de Producción.	Despliegue con éxito (Monitoreo de humo).

Nota: El equipo debe justificar por qué una etapa (Stage) debe funcionar como una **Gate (Puerta de Control)**, es decir, un punto donde el *pipeline* se detiene si no se cumplen ciertos criterios de calidad o seguridad.

3. Entregables del Proyecto

El equipo deberá entregar dos archivos:

Entregable 1: Diagrama de Arquitectura (Conceptual)

Cree un diagrama de flujo simple que represente su *pipeline* DevSecOps propuesto, siguiendo las 7 etapas de la sección C.

- Muestre claramente dónde se ejecuta cada herramienta de seguridad (SAST, DAST, SCA).
- Indique dónde están las puertas de control (Gates).

Entregable 2: Reporte de Justificación (Máx. 500 palabras)

Un reporte formal en formato escrito (Markdown o PDF) que cubra los puntos A, B y C del apartado 2, enfocándose en la justificación. El reporte debe responder específicamente a estas preguntas:

1. **¿Por qué eligieron poner SAST antes de DAST? (Principio Shift-Left)**
2. **¿Cómo garantiza su *pipeline* el requisito de alta seguridad y el requisito de velocidad al mismo tiempo?**
3. **¿Qué es lo más importante que debe proteger el *Hardening* del entorno en este caso (FastTicket)?**

4. Criterios de Evaluación

Su proyecto será evaluado en base a:

Criterio	Descripción	Puntos
Integración DevSecOps	Claridad en la ubicación de SAST, DAST y SCA en las etapas de **Code** y **Build** (Shift-Left).	30%
Aplicación de Hardening	Inclusión de tareas de *Hardening* adecuadas para contenedores, secretos y entorno de Prod.	25%
Lógica del Pipeline	La secuencia de las 7 etapas es lógica, eficiente y cada **Gate** está bien justificada.	25%

Claridad y Justificación	El reporte responde de manera clara y concisa a las preguntas clave del negocio.	20%
TOTAL		**100%**