Práctica Final Integrada – Despliegue Cloud-Native Seguro con IaC y Kubernetes

Peso: 30% de la nota final

Modalidad: Trabajo en grupo (4–5 personas)

La empresa ficticia **"CloudEdu Services"** desea migrar una aplicación web interna a una arquitectura cloud-native.
El equipo de DevOps (vosotros) deberá diseñar e implementar una solución que cumpla con los siguientes criterios:

1. La aplicación debe correr en **contenedores Docker**, gestionados por **Kubernetes**.
2. La infraestructura debe ser **reproducible** con Terraform o Ansible.
3. El sistema debe contemplar **volúmenes persistentes** y al menos un **servicio expuesto**.
4. Debe existir un **modelo de control de acceso (IAM)** que limite quién puede modificar o desplegar recursos.
5. Todo debe documentarse con **claridad técnica**, incluyendo diagramas, comandos y reflexión final.

Requisitos técnicos mínimos

Bloque	Requisito obligatorio	Herramientas recomendadas
Infraestructura (IaC)	Provisión automática de entorno (VMs, red o namespaces).	Terraform / Ansible
Contenedores	Imagen Docker personalizada (no usar solo nginx:latest).	Docker, Docker Hub
Orquestación	Despliegue y gestión con Kubernetes.	Minikube, EKS, AKS o GKE
Persistencia	Uso de volumes o PersistentVolumeClaim.	YAML Kubernetes
Red y exposición	Al menos un Service accesible desde el host o navegador.	NodePort / Ingress
Seguridad	Roles y políticas IAM coherentes con la arquitectura.	AWS Educate / Simulación local
Documentación	Instrucciones detalladas, diagrama y conclusiones.	Markdown / Word / PDF

Requisitos opcionales (mejoran la nota hasta +10%)

- Configuración de **pipeline CI/CD** (GitHub Actions, GitLab CI, Jenkins, etc.).
- Dashboard básico de **monitorización (Prometheus, Grafana, CloudWatch)**.
- Control de versiones del código en **GitHub o GitLab** con ramas organizadas.
- Implementación de **backup automatizado** o snapshots.

Entrega final

El grupo entregará un **paquete o repositorio** que incluya:

1. **Código fuente completo**
    - Dockerfiles, manifiestos YAML, ficheros Terraform/Ansible.
2. **Documentación técnica**
    - Descripción de la arquitectura y componentes.
    - Diagrama de red y dependencias.
    - Instrucciones de despliegue y configuración.
    - Políticas IAM aplicadas.
    - Principales problemas y cómo se resolvieron.
3. **Evidencia del funcionamiento**
    - Capturas de pantalla, logs o vídeo corto de demostración.
4. **Reflexión final (1 página máx.)**
    - Decisiones técnicas clave.
    - Dificultades encontradas.
    - Aprendizaje y roles del equipo.

Recomendaciones metodológicas

- Cada grupo debe **dividir roles** (arquitectura, IaC, orquestación, seguridad, documentación).
- Se recomienda trabajar con **control de versiones (Git)** desde el inicio.
- Las entregas parciales deben incluir **evidencias de progreso**.
- La **defensa oral** debe demostrar comprensión, no solo ejecución técnica.
- Se valorará la **coherencia entre el código y la documentación**.

Entrega y defensa

- **Formato:** ZIP o enlace a repositorio privado.
- **Plazo:** Semana 14 (día y hora según campus virtual).
- **Defensa:** 10–15 minutos por grupo, con demostración funcional.

Durante la defensa se valorará:

- Comprensión del diseño y decisiones técnicas.
- Capacidad de justificar configuraciones (por qué, no solo cómo).
- Trabajo coordinado y participación equilibrada.

Rúbrica de evaluación (30%)

Criterio	Descripción	Peso
Diseño arquitectónico	Claridad, coherencia, modularidad y escalabilidad.	20%
Implementación técnica (Docker + K8s + IaC)	Funcionamiento correcto, automatización y buenas prácticas.	30%
Seguridad y control de acceso	Aplicación correcta de IAM, aislamiento y permisos.	15%
Documentación técnica y claridad	Orden, precisión, justificación y diagrama.	15%
Trabajo en equipo y organización	Reparto de tareas, comunicación y coordinación.	10%
Defensa y presentación final	Claridad, dominio técnico y demostración práctica.	10%