

Dominik Heinrich, arquitecto de soluciones DevOps multicloud e IA

Reduce tu factura cloud sin romper nada

La checklist FinOps que uso en plataformas Azure, AWS y Kubernetes. Diez pasos, en el orden que más ahorra con el menor riesgo.



Ilustración: cómo baja normalmente una factura cloud en un proyecto FinOps

De un proyecto FinOps reciente, anonimizado

−48 %	**7 → 4**	**~40×**	**~35**
gasto en Azure en cinco semanas	nodos de Kubernetes	menos carga en Cosmos DB	cambios en un mes, un ticket cada uno

Los diez pasos

Recórrelos en orden y márcalos. Cada uno dice qué comprobar antes de cambiar nada.

01 Ver a dónde va el dinero

- [ ] Saca el coste amortizado por recurso y medidor desde la API de facturación, no solo la factura mensual.
- [ ] Construye una previsión del gasto para ver el efecto de un cambio en días, no a fin de mes.
- [ ] Etiqueta cada recurso con producto, equipo y entorno. El gasto sin etiqueta tiene responsable en una semana.
- [ ] Envía un breve informe diario de costes al chat del equipo. Lo que se ve, se arregla.

02 Comprobar antes de recortar

- [ ] **Para cada candidato:** qué corre ahí, quién lo usa y cuál es la carga pico (p95), nunca la media.
- [ ] Todo lo que un cliente pueda notar queda fuera hasta demostrar que es seguro.
- [ ] Un cambio por ticket, cada uno con su impacto, su riesgo y su forma de revertirlo.

03 Dimensionar bien el cómputo

- [ ] Fija los requests y limits de Kubernetes según el uso p95 real y deja que el número de nodos se ajuste.
- [ ] Usa autoescalado (cluster autoscaler o Karpenter) y capacidad spot para los trabajos que se pueden interrumpir.
- [ ] **En un proyecto:** un clúster de producción pasó de 7 a 4 nodos sin impacto para los clientes.

04 Reducir y eliminar

- [ ] VMs ociosas, discos y volúmenes huérfanos, IPs sin asignar, snapshots antiguos y recursos de prueba olvidados.
- [ ] Réplicas y alta disponibilidad de bases de datos solo donde el negocio de verdad las necesita.

05 Escribir una política de retención de datos

- [ ] **Logs:** debug desactivado en producción, retención por tipo de log y almacenamiento barato para los logs de auditoría.
- [ ] **Copias de seguridad:** retención según tus objetivos de recuperación y obligaciones legales, no según los valores por defecto.
- [ ] **Almacenamiento de objetos:** reglas de ciclo de vida que pasan los datos antiguos a niveles fríos y borran lo que nadie necesita.

06 Arreglar las consultas caras

- [ ] Busca las consultas que más cuestan, no las más frecuentes. Una sola corrección redujo unas 40 veces la carga de una base de datos Cosmos DB.
- [ ] Mueve las bases de datos pequeñas de clientes al clúster (por ejemplo con CloudNativePG) cuando compense, comprobando recuentos y checksums.

07 Comprometerse al final, no al principio

- [ ] Compra reservas o savings plans después del ajuste de tamaño, para no comprometerte con máquinas que vas a quitar.
- [ ] Trata los compromisos como difíciles de deshacer: los savings plans de Azure no se pueden cancelar, los Savings Plans de AWS solo se pueden devolver en siete días y los reembolsos de reservas de Azure tienen un límite. Cuando cambie el entorno, usa las opciones de intercambio que existan.

08 Apagar los entornos

- [ ] Dev y test apagados por defecto de noche y los fines de semana, y reconstruidos automáticamente por la mañana.
- [ ] Entornos de preview con caducidad y limpiezas nocturnas de recursos huérfanos.

09 Vigilar la red

- [ ] Alertas sobre el tráfico entre zonas y de salida, y revisión de NAT gateways y balanceadores sin responsable.

10 Mantenerlo bajo

- [ ] Presupuestos con alertas de previsión por equipo y un informe semanal de una página con previsión a dos meses para la dirección.
- [ ] Un responsable por centro de coste, y la checklist repasada cada trimestre.

Los errores que más veo

- Comprar reservas primero y dimensionar después, pagando máquinas que ya no se usan.
- Dimensionar según la media: el clúster parece vacío hasta el pico del lunes.
- Borrar antes de comprobar quién lo usa. Un consumidor olvidado cuesta más que el ahorro.
- Apagar los logs en vez de moverlos a niveles más baratos, y perder la trazabilidad.
- Un gran fin de semana de limpieza en vez de un ticket por cambio, y nadie sabe qué rompió qué.

Cómo podemos trabajar juntos

La misma calidad para una startup de diez personas y para un grupo global. El formato se adapta a ti.

Startups

Responsable de DevOps e IA a tiempo parcial

Unos días al mes: arquitectura, pipeline de releases y funciones de IA, para que los fundadores se centren en los clientes. Del prototipo a producción sin reescribir.

Pymes

Un proyecto con un resultado claro

Un objetivo definido, como una factura cloud más baja, una migración o los primeros agentes de IA en producción, con alcance cerrado, avances semanales y una entrega que tu equipo pueda operar.

Grandes empresas

Un arquitecto dentro de tus equipos

Integrado con tus equipos de plataforma, seguridad y producto: decisiones de arquitectura, gobernanza de la IA y entornos regulados. Para alcances mayores traigo un equipo senior de confianza.

Cómo trabajo

Decisiones por escrito: registros de decisiones de arquitectura con sus pros y contras, para que tu equipo sepa el porqué, no solo el qué.

Operado, no solo entregado: SLOs, alertas sobre resultados de negocio y una rutina de incidencias.

El conocimiento se queda contigo: runbooks, documentación y talleres para tu equipo.

Seguro por defecto: OIDC e identidades gestionadas, mínimo privilegio, secretos en un almacén seguro y trazabilidad.

Medido: coste, tiempo de entrega, disponibilidad y horas ahorradas, con un informe semanal.

Un único interlocutor que habla con ingeniería y con dirección, en alemán, inglés y español.



¿Quieres que lo aplique a tu factura?

Reservar una llamada de 30 minutos

E-mail mail@dominikheinrich.ch
Web dominikheinrich.ch

Disponible en todo el mundo en remoto. Presencial bajo petición, según el país.