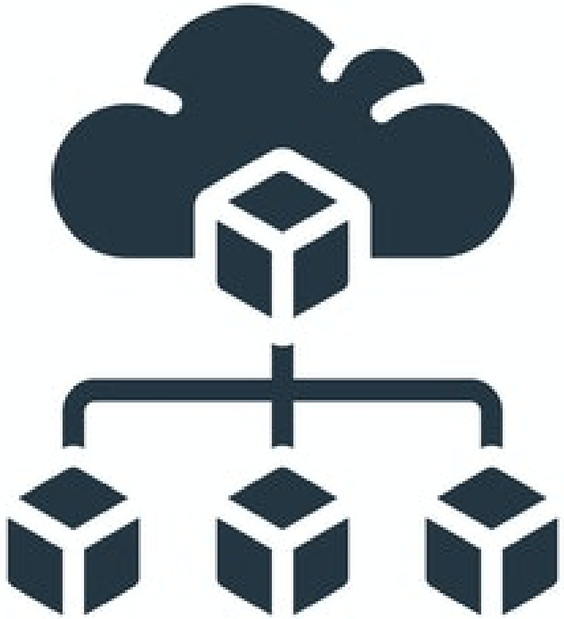
APLICACIÓN DE TODO'S

Ingenieros DevOps Santiago Valencia García y Danna Valentina López Muñoz



Tarea	Estado	Responsable	Fecha límite	Prioridad
Definir estrategia de branching para de	Completado	Valentina Lopez Muñoz	16 de septiembre de 2025	Alta
Definir estrategia de branching para op	Completado	Valentina Lopez Muñoz	16 de septiembre de 2025	Alta
Creación de Dockerfile para cada micro	Completado	Santiago Valencia	18 de septiembre de 2025	Alta
Escribir script de Terraform	Completado	Valentina Lopez Muñoz Santiago Valenc	28 de septiembre de 2025	Alta
Crear un workflow de GitHub Actions (i	Completado	Santiago Valencia	28 de septiembre de 2025	Alta
Creación de pipelines para desarrollo	Completado	Valentina Lopez Muñoz Santiago Valenc	28 de septiembre de 2025	Alta
Creación de pipelines para infraestructu	Completado	Valentina Lopez Muñoz Santiago Valenc	28 de septiembre de 2025	Alta
Crear secretos en GitHub Actions para l	Completado	Santiago Valencia	15 de septiembre de 2025	Media
Implementar patrones	Completado	Valentina Lopez Muñoz Santiago Valenc	28 de septiembre de 2025	Media
Actualizar el diagrama de arquitectura	Completado	Valentina Lopez Muñoz Santiago Valenc	28 de septiembre de 2025 → 28	Baja
Realizar documentación	Completado	Valentina Lopez Muñoz Santiago Valenc	28 de septiembre de 2025	Baja

ESTRATEGIA DE BRANCHING

Se utilizo un GitHub Flow unificado donde la rama main permanece protegida y en estado de producción. Todo el trabajo se realiza en ramas de corta duración que siguen una convención de nomenclatura estructurada: <tipo>/<equipo>/<descripción>, donde el tipo puede ser `feature`, `fix`, `chore` o `hotfix`, el equipo se distingue entre dev (para desarrollo) y ops (infraestructura y operaciones), y la descripción es una breve explicación del trabajo realizado.

¿QUÉ SE HIZO?

1. Creación de la Infraestructura
2. Dockerización de los microservicios
3. Implementación de patrones
   -Cache-aside
   -Circuit breaker
4. Despliegue en Azure
5. Automatización de cambios y despliegue

TECNOLOGÍAS USADAS

- Lenguajes y Frameworks: Go, Java (Spring Boot), Node.js (Express), Python y Vue.js.
- DevOps e Infraestructura: Docker, Terraform, Ansible y GitHub Actions, todo desplegado en Azure.
- Bases de Datos y Caché: Redis y H2 Database.
- Otras Herramientas: Nginx como servidor web, y Zipkin para el seguimiento distribuido.

DIAGRAMA PREVIO DE LA ARQUITECTURA

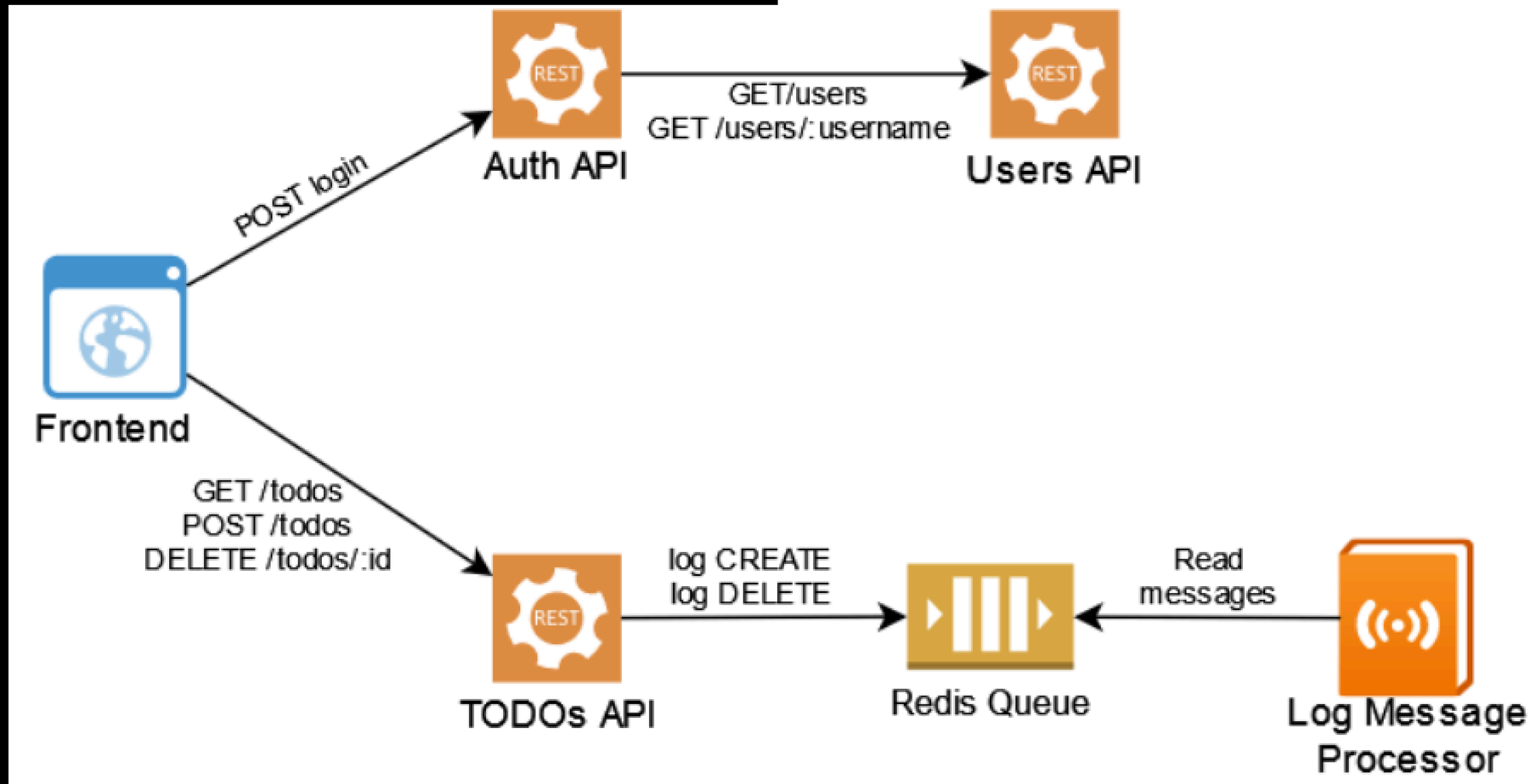
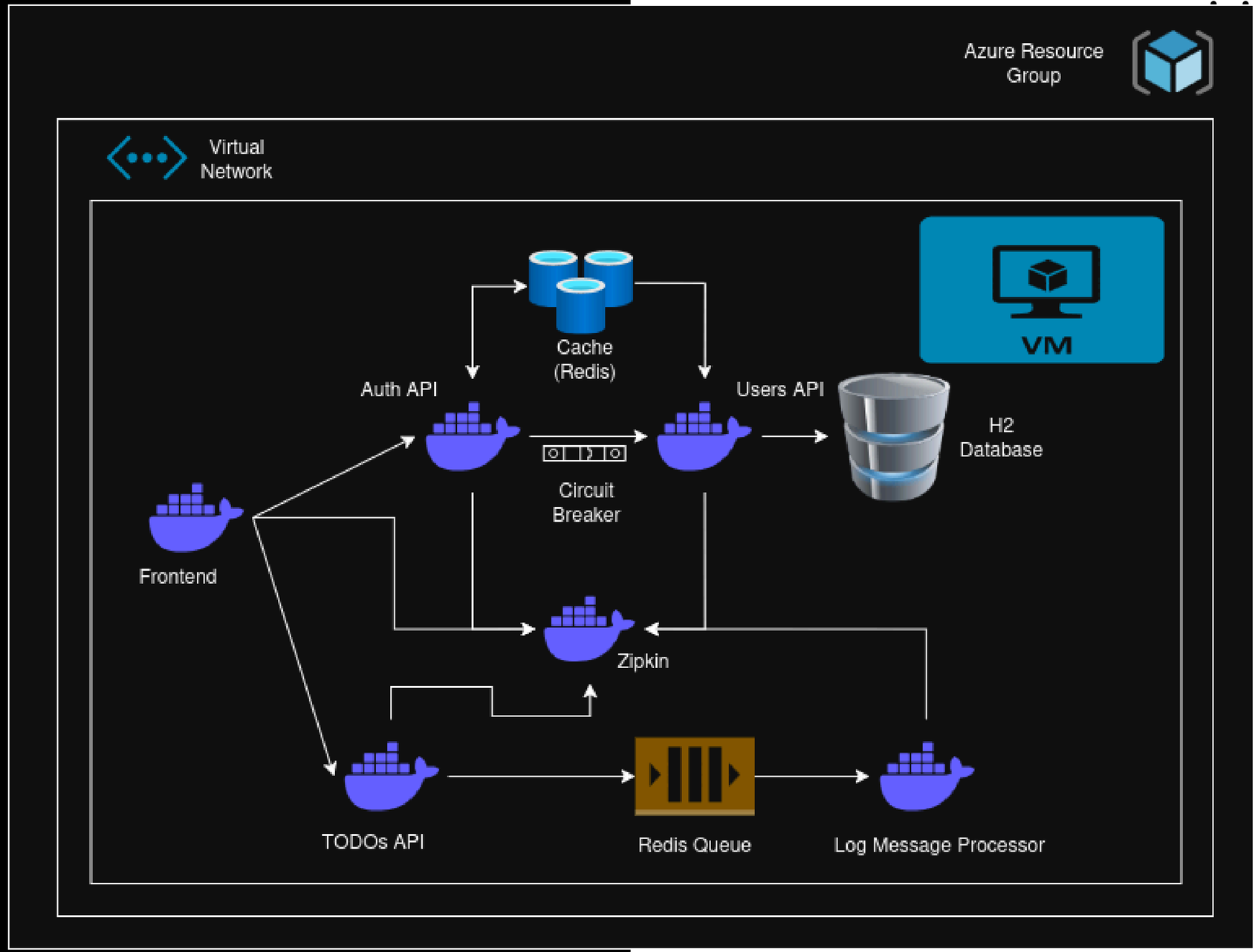


DIAGRAMA ACTUAL DE LA ARQUITECTURA



VENTAJAS DE LA SOLUCIÓN

1

Despliegue Automático

Un simple git push despliega automáticamente toda la infraestructura y aplicaciones

2

Resiliencia a Fallos

El Circuit Breaker previene que un servicio caído afecte todo el sistema

3

Performance Optimizada

Cache-Aside reduce la latencia de consultas frecuentes

4

Escalabilidad Independiente

Cada servicio puede escalarse por separado según su demanda

5

Despliegue Granular

Cada microservicio se despliega independientemente

6

Costos Optimizados

Solo se paga por los recursos que realmente se usan en Azure

ANÁLISIS DE COSTOS

Componente	Especificación Clave (desde Terraform)	Tarifa Oficial (Azure East US)	Cálculo Mensual (730h/mes)	Estimación Mensual
Máquina Virtual	`Standard_B2s` (2 vCPU, 4 GiB RAM)	$0.0342 / hora	`$0.0342 × 730h`	**~$25.00**
Disco del Sistema	`32 GiB` (Standard HDD LRS)	$1.54 / mes	*(Tarifa mensual directa)*	**~$1.54**
IP Pública	`Standard SKU`, Estática	$0.005 / hora	`$0.005 × 730h`	**~$3.65**
Transferencia de Datos	Saliente	*(Primeros 100GB/mes gratis)*	Variable	**$0.00***
			Total Estimado	**~$30.19**

DEMOSTRACIÓN

¡Gracias por su atención!