

UNIVERSIDAD PRIVADA BOLIVIANA
LA PAZ
FACULTAD DE INGENIERÍA Y ARQUITECTURA



Proyecto Tercer Parcial

Implementación de Ciclo DevOps y Observabilidad

Estudiantes:

Ivan Iver Poma Maidana - 70419
Wendy Evelyn Cáceres Vasquez - 68874
Andres Raul Sanchez Andrade - 69017

Docente: Rayner Mendieta Villalba

La Paz – Bolivia – 18 de Diciembre de 2025

Índice

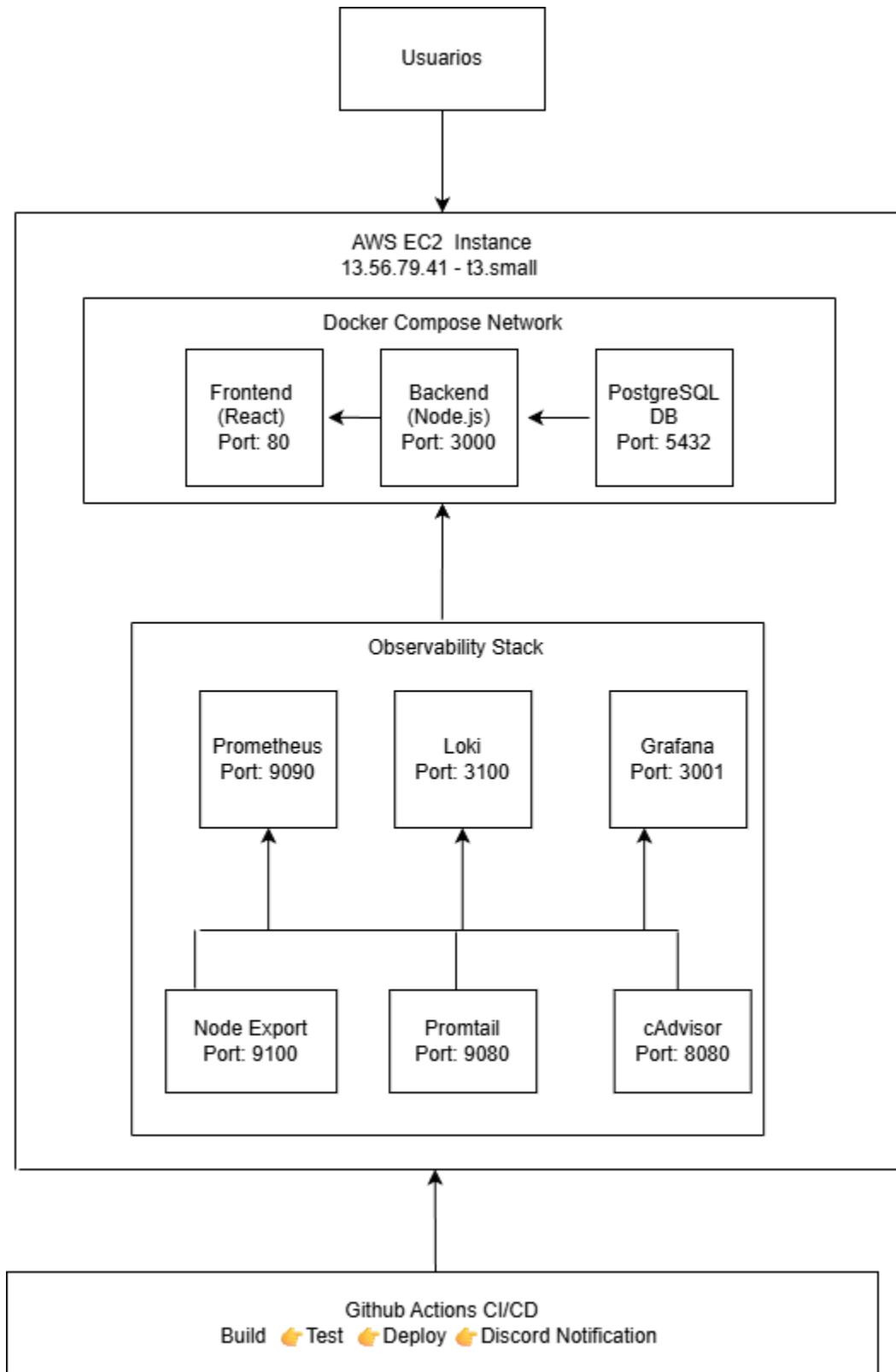
1. Requisitos y alcance.....	3
2. Arquitectura.....	5
3. Dockerfiles & Archivos importantes.....	7
4. CI/CD.....	13
5. Loki config.....	17
6. Prometheus config.....	18
7. Despliegue en EC2.....	19
8. Seguridad y secretos.....	22
9. Pruebas.....	22
10. Operación y mantenimiento.....	29
11. Conclusiones.....	30

1. Requisitos y alcance

Característica	Estado
Frontend SPA con CRUD completo	Implementado
Backend API REST con lógica de negocio y endpoints para la SPA	Implementado
Base de datos relacional con persistencia	Implementado
Cada componente en contenedor Docker independiente	Implementado
Dockerfiles reproducibles y buenas prácticas	Implementado
docker-compose.yml	Implementado
Despliegue final de los contenedores en una instancia EC2	Implementado
Pipeline CI/CD en GitHub Action	Implementado
Gestión de secretos mediante GitHub Secrets	Implementado
Monitoreo de Infraestructura (Prometheus + Node Exporter / cAdvisor)	Implementado

README con instrucciones de ejecución local y URL pública de la aplicación	Implementado
Centralización y visualización (Loki + Promtail)	Implementado
Auto-restart	Implementado
Dashboard unificado en Grafana	Implementado
Notificaciones Discord	Implementado
Monitoreo visible en Grafana	Implementado
Rollback automatizado	No Implementado
TLS/HTTPS	No Implementado
Escalado horizontal simple	No implementado

2. Arquitectura



La aplicación se despliega en una única instancia AWS EC2 (t3.small – IP pública: 13.56.79.41) utilizando Docker Compose para orquestar todos los servicios en una red interna dedicada.

El flujo principal es el siguiente:

1. Acceso del usuario: Cualquier usuario accede desde su navegador a la URL pública de la aplicación <http://13.56.79.41> o en algún apartado específico.
2. El tráfico ingresa a AWS a través del Security Group configurado, que solo permite:
 - Puerto 22 (SSH) → IPs autorizadas
 - Puerto 80 → Aplicación web (público)
 - Puerto 3001 → Grafana (público, para evaluación y monitoreo)
3. Dentro de la instancia EC2 se ejecutan todos los contenedores orquestados por `docker-compose.yml`:
 - Frontend (React + Nginx) → puerto 80 → sirve la SPA
 - Backend (Node.js + Express) → puerto 3000 → expone la API REST
 - PostgreSQL → puerto 5432 (solo interno) → persistencia con volumen dedicado
4. El Frontend consume el Backend mediante peticiones Axios.
5. El Backend se conecta a PostgreSQL usando variables de entorno seguras.

Adicionalmente, se integra un stack completo de observabilidad (Grafana Stack) dentro de la misma instancia:

- Node Exporter (puerto 9100) → expone métricas del sistema operativo y hardware del EC2 (CPU, RAM, disco).
- cAdvisor (puerto 8080) → monitorea métricas detalladas de todos los contenedores Docker.
- Prometheus (puerto 9090) → scrapea automáticamente métricas de Node Exporter y cAdvisor.
- Promtail (puerto 9080) → recolecta logs de todos los contenedores y del host.
- Loki (puerto 3100) → almacena y indexa los logs recolectados por Promtail.
- Grafana (puerto 3001) → visualización unificada: dashboards de métricas (Prometheus) y explorador de logs (Loki). Incluye dashboards preconfigurados para infraestructura y aplicación.

Finalmente, el pipeline CI/CD se activa desde un solo repositorio donde ahí se encuentran los dos repositorios tanto del Frontend como del Backend:

- <https://github.com/ObiTobi64/casa-cambios-devops>

Cada push a main desencadena automáticamente: build de imágenes → push a registry → despliegue en EC2 vía SSH → docker compose up -d → notificación automática a Discord con detalles del commit y autor.

3. Dockerfiles & Archivos importantes

Docker-compose.yml:

Base de datos PostgreSQL

```
1  version: '3.8'
2
3  services:
4      # Base de datos PostgreSQL
5      postgres:
6          image: postgres:15-alpine
7          container_name: casa-cambios-db
8          restart: unless-stopped
9          environment:
10             POSTGRES_USER: ${DB_USER:-postgres}
11             POSTGRES_PASSWORD: ${DB_PASSWORD:-postgres}
12             POSTGRES_DB: ${DB_NAME:-casacambios}
13          ports:
14              - "5432:5432"
15          volumes:
16              - postgres_data:/var/lib/postgresql/data
17          networks:
18              - app-network
19          healthcheck:
20              test: ["CMD-SHELL", "pg_isready -U postgres"]
21              interval: 10s
22              timeout: 5s
23              retries: 5
24
```

Backend Api

```
55      # Frontend React
56      frontend:
57          image: casa-cambios-frontend:latest
58          container_name: casa-cambios-frontend
59          restart: unless-stopped
60          ports:
61              - "80:80"
62          depends_on:
63              - backend
64          networks:
65              - app-network
66          logging:
67              driver: "json-file"
68          options:
69              max-size: "10m"
70              max-file: "3"
71          labels: "service=frontend"

25      # Backend API
26      backend:
27          image: casa-cambios-backend:latest
28          container_name: casa-cambios-backend
29          restart: unless-stopped
30          ports:
31              - "3000:3000"
32          environment:
33              NODE_ENV: development
34              DATABASE_URL: postgresql://${DB_USER:-postgres}:${DB_PASSWORD:-postgres}@postgres:5432/${DB_NAME:-casacambios}
35              DB_HOST: postgres
36              DB_PORT: 5432
37              DB_USER: ${DB_USER:-postgres}
38              DB_PASSWORD: ${DB_PASSWORD:-postgres}
39              DB_NAME: ${DB_NAME:-casacambios}
40              PORT: 3000
41          depends_on:
42              postgres:
43                  condition: service_healthy
44          volumes:
45              - ./proyecto2-back/logs:/app/logs
46          networks:
47              - app-network
48          logging:
49              driver: "json-file"
50              options:
51                  max-size: "10m"
52                  max-file: "3"
53          labels: "service=backend"
```

Frontend React

Prometheus

```
73      # Prometheus - Recolector de métricas
74
75      prometheus:
76          image: prom/prometheus:latest
77          container_name: prometheus
78          restart: unless-stopped
79          ports:
80              - "9090:9090"
81          volumes:
82              - ./monitoring/prometheus/prometheus.yml:/etc/prometheus/prometheus.yml
83              - prometheus_data:/prometheus
84          command:
85              - '--config.file=/etc/prometheus/prometheus.yml'
86              - '--storage.tsdb.path=/prometheus'
87              - '--storage.tsdb.retention.time=15d'
88              - '--web.console.libraries=/etc/prometheus/console_libraries'
89              - '--web.console.templates=/etc/prometheus/consoles'
90              - '--web.enable-lifecycle'
91          networks:
92              - app-network
```

EC2

```
93      # Node Exporter - Métricas del sistema EC2
94
95      node-exporter:
96          image: prom/node-exporter:latest
97          container_name: node-exporter
98          restart: unless-stopped
99          ports:
100              - "9100:9100"
101          command:
102              - '--path.procfs=/host/proc'
103              - '--path.rootfs=/rootfs'
104              - '--path.sysfs=/host/sys'
105              - '--collector.filesystem.mount-points-exclude=^/(sys|proc|dev|host|etc)(\$\$|/)'
106          volumes:
107              - /proc:/host/proc:ro
108              - /sys:/host/sys:ro
109              - /:/rootfs:ro
110          networks:
111              - app-network
```

cAdvisor

```
112      # cAdvisor - Métricas de contenedores Docker
113      cadvisor:
114          image: gcr.io/cadvisor/cadvisor:latest
115          container_name: cadvisor
116          restart: unless-stopped
117          ports:
118              - "8080:8080"
119          volumes:
120              - /:/rootfs:ro
121              - /var/run:/var/run:ro
122              - /sys:/sys:ro
123              - /var/lib/docker:/var/lib/docker:ro
124              - /dev/disk/:/dev/disk:ro
125          privileged: true
126          devices:
127              - /dev/kmsg
128          networks:
129              - app-network
```

Loki

```
131      # Loki - Servidor de logs
132      loki:
133          image: grafana/loki:latest
134          container_name: loki
135          restart: unless-stopped
136          ports:
137              - "3100:3100"
138          volumes:
139              - ./monitoring/loki/loki-config.yml:/etc/loki/local-config.yaml
140              - loki_data:/loki
141          command: -config.file=/etc/loki/local-config.yaml
142          networks:
143              - app-network
```

Promtail

```
145      # Promtail - Recolector de logs
146      promtail:
147          image: grafana/promtail:latest
148          container_name: promtail
149          restart: unless-stopped
150          volumes:
151              - ./monitoring/promtail/promtail-config.yml:/etc/promtail/config.yml
152              - /var/lib/docker/containers:/var/lib/docker/containers:ro
153              - /var/run/docker.sock:/var/run/docker.sock
154              - ./proyecto2-back/logs:/var/log/backend:ro
155          command: -config.file=/etc/promtail/config.yml
156          depends_on:
157              - loki
158          networks:
159              - app-network
```

Grafana

```
161      # Grafana - Visualización de métricas y logs
162      grafana:
163          image: grafana/grafana:latest
164          container_name: grafana
165          restart: unless-stopped
166          ports:
167              - "3001:3000"
168          environment:
169              GF_SECURITY_ADMIN_USER: ${GRAFANA_USER:-admin}
170              GF_SECURITY_ADMIN_PASSWORD: ${GRAFANA_PASSWORD:-admin123}
171              GF_INSTALL_PLUGINS: grafana-clock-panel,grafana-simple-json-datasource
172              GF_SERVER_ROOT_URL: http://13.56.79.41:3001
173          volumes:
174              - grafana_data:/var/lib/grafana
175              - ./monitoring/grafana/provisioning:/etc/grafana/provisioning
176              - ./monitoring/grafana/dashboards:/var/lib/grafana/dashboards
177          depends_on:
178              - prometheus
179              - loki
180          networks:
181              - app-network
182
183      networks:
184          app-network:
185              driver: bridge
186
187      volumes:
188          postgres_data:
189          prometheus_data:
190          loki_data:
191          grafana_data:
```

- **Imágenes ligeras:** Uso de variantes :alpine (PostgreSQL) y imágenes oficiales optimizadas del stack Grafana (Prometheus, Loki, Promtail, Grafana), minimizando consumo de RAM y disco en la instancia t3.small.
- **Persistencia crítica:** Volúmenes dedicados (postgres_data, prometheus_data, loki_data, grafana_data) garantizan que datos de la aplicación, series temporales, logs indexados y dashboards/configuraciones de Grafana persistan ante reinicios o redeployes.
- **Healthchecks y dependencias inteligentes:** PostgreSQL con healthcheck (pg_isready) y depends_on: service_healthy en backend, evitando arranques prematuros. Grafana espera a Prometheus y Loki.
- **Logging controlado y centralizado:** Driver json-file con rotación automática (10 MB × 3 archivos) en frontend/backend para evitar saturación, mientras Promtail accede a /var/lib/docker/containers y al socket Docker para recolectar logs en tiempo real y enviarlos a Loki.
- **Política de reinicio robusta:** restart: unless-stopped en todos los servicios, asegurando alta disponibilidad automática.
- **Red única y segura:** Todos los servicios en app-network (bridge), permitiendo comunicación interna sin exponer puertos innecesarios al exterior.

Dockerfile - Backend

```
proyecto2-back > 🐳 Dockerfile > ...
AndresSanchez, last week | 1 author (AndresSanchez)
1 FROM node:18-alpine (last pushed 8 months ago)
2
3 WORKDIR /app
4
5 COPY package*.json .
6
7 RUN npm ci --omit=dev
8
9 COPY . .
10
11 EXPOSE 3000
12
13 CMD ["npm", "start"]
14
```

Dockerfile - Frontend

```

proyecto2-front > 🐳 Dockerfile > ...
1  FROM node:20-alpine AS build (last pushed 13 hours ago)
2  WORKDIR /app
3  COPY package*.json ./
4  RUN npm ci
5  COPY . .
6
7
8  ARG VITE_API_URL
9  ENV VITE_API_URL=$VITE_API_URL
10
11 RUN npm run build
12
13 # Production stage - Usando Apache (http)
14 FROM httpd:alpine (last pushed 8 hours ago)
15
16 # Copiar los archivos construidos al directorio de Apache
17 COPY --from=build /app/dist /usr/local/apache2/htdocs/
18
19 # Habilitar mod_rewrite para que funcionen las rutas de React
20 RUN sed -i '/LoadModule rewrite_module/s/^#//g' /usr/local/apache2/conf/httpd.conf
21
22 # Configurar FallbackResource para SPA (La forma más simple en Apache moderno)
23 # Esto le dice a Apache: "Si no encuentras el archivo, sirve index.html"
24 RUN echo "FallbackResource /index.html" >> /usr/local/apache2/conf/httpd.conf
25
26 EXPOSE 80

```

4. CI/CD

BaCI/CD – .github/workflows/ci-cd.yml

```

1   name: CI/CD Pipeline - Casa de Cambio
2
3   on:
4     push:
5       branches:
6         - main
7         - master
8     pull_request:
9       branches:
10        - main
11        - master
12     workflow_dispatch:
13
14   env:
15     EC2_HOST: ${{ secrets.EC2_HOST }}
16     EC2_USER: ${{ secrets.EC2_USER }}
17
18   jobs:
19     build-and-push:
20       name: Build and Push to Docker Hub
21       runs-on: ubuntu-latest
22
23       steps:
24         - name: Checkout code
25           uses: actions/checkout@v4
26
27         - name: Clone Frontend Repository
28           run: |
29             git clone https://github.com/certi-devops-team/proyecto2-front.git
30
31         - name: Clone Backend Repository
32           run: |
33             git clone https://github.com/certi-devops-team/proyecto2-back.git
34
35         - name: Login to Docker Hub
36           uses: docker/login-action@v3
37           with:

```

```

37     with:
38       username: ${{ secrets.DOCKER_USERNAME }}
39       password: ${{ secrets.DOCKER_PASSWORD }}
40
41   - name: Build and Push Backend
42     uses: docker/build-push-action@v5
43     with:
44       context: ./proyecto2-back
45       push: true
46       tags: ${{ secrets.DOCKER_USERNAME }}/casa-cambios-backend:latest
47
48   - name: Build and Push Frontend
49     uses: docker/build-push-action@v5
50     with:
51       context: ./proyecto2-front
52       push: true
53       build-args: |
54         VITE_API_URL=http://${{ env.EC2_HOST }}:3000
55       tags: ${{ secrets.DOCKER_USERNAME }}/casa-cambios-frontend:latest
56
57 deploy:
58   name: Deploy to EC2
59   runs-on: ubuntu-latest
60   needs: build-and-push
61   if: github.ref == 'refs/heads/main' || github.ref == 'refs/heads/master'
62
63 steps:
64   - name: Checkout code
65     uses: actions/checkout@v4
66
67   - name: Configure SSH
68     run: |
69       mkdir -p ~/.ssh
70       echo "${{ secrets.EC2_SSH_KEY }}" > ~/.ssh/id_rsa
71       chmod 600 ~/.ssh/id_rsa
72       ssh-keyscan -H ${{ env.EC2_HOST }} >> ~/.ssh/known_hosts

72       ssh-keyscan -H ${{ env.EC2_HOST }} >> ~/.ssh/known_hosts
73
74   - name: Copy files to EC2
75     run: |
76       scp -i ~/.ssh/id_rsa docker-compose.yml ${{ env.EC2_USER }}@${{ env.EC2_HOST }}:~/casa-cambios-devops/
77       scp -i ~/.ssh/id_rsa -r monitoring ${{ env.EC2_USER }}@${{ env.EC2_HOST }}:~/casa-cambios-devops/
78
79   - name: Deploy to EC2
80     run: |
81       ssh -i ~/.ssh/id_rsa ${{ env.EC2_USER }}@${{ env.EC2_HOST }} << 'EOF'
82         # Ir al directorio del proyecto
83         cd ~/casa-cambios-devops
84
85         # Crear archivo .env si no existe
86         if [ ! -f .env ]; then
87             cat > .env << 'ENVEOF'
88             DB_USER=postgres
89             DB_PASSWORD=postgres123
90             DB_NAME=casacambios
91             GRAFANA_USER=admin
92             GRAFANA_PASSWORD=admin123
93             ENVEOF
94         fi
95
96         # Exportar variables de imagen para docker-compose
97         export BACKEND_IMAGE=${{ secrets.DOCKER_USERNAME }}/casa-cambios-backend:latest
98         export FRONTEND_IMAGE=${{ secrets.DOCKER_USERNAME }}/casa-cambios-frontend:latest
99
100        # Pull de las nuevas imágenes
101        docker-compose pull
102
103        # Reiniciar servicios con las nuevas imágenes
104        docker-compose up -d
105
106        # Limpiar imágenes antiguas
107        docker image prune -f

```

```

108      EOF
109
110      - name: Verify Deployment
111        run: |
112          ssh -i ./ssh/id_rsa ${{ env.EC2_USER }}@${{ env.EC2_HOST }} << 'EOF'
113          cd ~/casa-cambios-devops
114          echo "Verificando servicios..."
115          sleep 10
116          docker-compose ps
117
118          # Verificar que los servicios estén corriendo
119          if [ $(docker-compose ps | grep -c "Up") -lt 5 ]; then
120              echo "Error: No todos los servicios están corriendo"
121              docker-compose logs --tail=50
122              exit 1
123          fi
124
125          echo "Todos los servicios están corriendo correctamente"
126
127
128      - name: Send Discord Notification - Success
129        if: success()
130        run: |
131          curl -H "Content-Type: application/json" \
132            -X POST \
133            -d '{
134              "embeds": [
135                {
136                  "title": "✅ Despliegue Exitoso - Casa de Cambios",
137                  "description": "El despliegue se completó satisfactoriamente",
138                  "color": #666999,
139                  "fields": [
140                    {
141                      "name": "Autor",
142                      "value": "${{ github.actor }}",
143                      "inline": true
144                    },
145                    {
146                      "name": "Rama",
147                      "value": "${{ github.ref_name }}",
148                      "inline": true
149                    },
150                    {
151                      "name": "Mensaje del Commit",
152                      "value": "${{ git log -1 --pretty=%B }}",
153                      "inline": false
154                    },
155                    {
156                      "name": "URLs",
157                      "value": "*${{ App:** http://${{ secrets.EC2_HOST }}** }}\n${{ Grafana:** http://${{ secrets.EC2_HOST }}**:3001\n**Prometheus:** http://${{ secrets.EC2_HOST }}** }}",
158                      "inline": false
159                    }
160                  ],
161                  "timestamp": "${{ date -u +%Y-%m-%dTH:%M:%S.000Z }}",
162                  "footer": {
163                    "text": "CI/CD Pipeline"
164                  }
165                }
166              ]
167            }' \
168            ${{ secrets.DISCORD_WEBHOOK_URL }}
169
170
171
172      - name: Send Discord Notification - Failure
173        if: failure()
174        run: |
175          curl -H "Content-Type: application/json" \
176            -X POST \
177            -d '{
178              "embeds": [
179
180                {
181                  "title": "🔴 Fallo en el Despliegue - Casa de Cambios",
182                  "description": "El despliegue falló. Revisa los logs para más detalles.",
183                  "color": #A51832,
184                  "fields": [
185                    {
186                      "name": "Commit",
187                      "value": "${{ git log -1 --pretty=%B }}",
188                      "inline": true
189                    },
190                    {
191                      "name": "Autor",
192                      "value": "${{ github.actor }}",
193                      "inline": true
194                    },
195                    {
196                      "name": "Rama",
197                      "value": "${{ github.ref_name }}",
198                      "inline": true
199                    },
200                    {
201                      "name": "Ver Logs",
202                      "value": "[GitHub Actions](https://github.com/${{ github.repository }}/actions/runs/${{ github.run_id }})",
203                      "inline": false
204                    }
205                ],
206                "timestamp": "${{ date -u +%Y-%m-%dTH:%M:%S.000Z }}",
207                "footer": {
208                  "text": "CI/CD Pipeline"
209                }
210              ]
211            }' \
212            ${{ secrets.DISCORD_WEBHOOK_URL }}

```

En nuestro archivo ci-cd.yml tenemos lo siguiente:

- Triggers flexibles: Se activa en push/PR a main/master o manualmente (workflow_dispatch).
- Clonación de repositorios: Descarga automáticamente frontend y backend desde sus repos independientes.
- Construye y sube imágenes a Docker Hub: Login seguro, build con Dockerfiles optimizados y push con tag latest (inyectando VITE_API_URL dinámica en frontend).
- Despliegue condicional: Solo en ramas main/master.
- SSH seguro a EC2: Configura clave privada desde secrets y agrega host a known_hosts.
- Copia de archivos críticos: Transfiere docker-compose.yml y carpeta monitoring completa al servidor.
- Preparación en EC2: Crea .env por defecto si no existe (credenciales seguras).
- Pull y redeploy inteligente: docker-compose pull + up -d (zero downtime) + image prune para limpiar disco.
- Verificación automática de salud: Espera 10s, revisa servicios "Up" y muestra logs si falla → workflow marca error.
- Notificaciones avanzadas a Discord: Embed rico con éxito (verde) o fallo (rojo), incluyendo SHA, autor, rama, mensaje commit, URLs (app, Grafana, Prometheus) y enlace directo a logs de GitHub Actions.
- Seguridad total: Todo secrets desde GitHub Secrets.

Secrets utilizados:

- DISCORD_WEBHOOK_URL
- DOCKER_PASSWORD
- DOCKER_USERNAME
- EC2_HOST
- EC2_SSH_KEY
- EC2_USER

5. Loki config

```
1 auth_enabled: false
2
3 server:
4   http_listen_port: 3100
5   grpc_listen_port: 9096
6
7 common:
8   instance_addr: 127.0.0.1
9   path_prefix: /loki
10  storage:
11    filesystem:
12      chunks_directory: /loki/chunks
13      rules_directory: /loki/rules
14      replication_factor: 1
15  ring:
16    kvstore:
17      store: inmemory
18
19 query_range:
20   results_cache:
21     cache:
22       embedded_cache:
23         enabled: true
24         max_size_mb: 100
25
26 schema_config:
27   configs:
28     - from: 2020-10-24
29       store: tsdb
30       object_store: filesystem
31       schema: v13
32       index:
33         prefix: index_
34         period: 24h
35
36 # Límites para evitar que Loki consuma demasiada memoria
37 limits_config:
38   reject_old_samples: true
39   reject_old_samples_max_age: 168h
40   ingestion_rate_mb: 10
41   ingestion_burst_size_mb: 20
42   per_stream_rate_limit: 5MB
43   per_stream_rate_limit_burst: 15MB
44   retention_period: 168h
45
46 # Configuración de compactación
47 compactor:
48   working_directory: /loki/compactor
49   retention_enabled: true
50   retention_delete_delay: 2h
51   retention_delete_worker_count: 150
52   delete_request_store: filesystem
```

- Modo standalone simple: auth_enabled: false (sin autenticación) y almacenamiento local en filesystem (/loki/chunks y /loki/rules). Ideal para instancias pequeñas como t3.small.
- Puertos expuestos:
 - HTTP: 3100 (ingestión y queries)
 - gRPC: 9096
- Esquema moderno: Usa TSDB (v13 desde 2020) con índices diarios (period: 24h) → más eficiente que versiones antiguas.
- Caché de resultados: Activada (embedded_cache, 100 MB max) → acelera consultas repetidas en Grafana.
- Límites de ingestión y retención
- Compactación y retención activa:

- Retención habilitada con borrado automático
- Directorio de trabajo: /loki/compactor
- Delay de 2h antes de borrar
- 150 workers para procesamiento rápido

6. Prometheus config

```

1   global:
2     scrape_interval: 15s
3     evaluation_interval: 15s
4     external_labels:
5       monitor: 'casa-cambios-monitor'
6
7   # Configuración de Alertmanager (opcional)
8   # alerting:
9   #   alertmanagers:
10  #     - static_configs:
11  #       - targets: ['alertmanager:9093']
12
13 # Reglas de alertas (opcional)
14 # rule_files:
15 #   - "alerts.yml"
16
17 scrape_configs:
18   # Métricas de Prometheus mismo
19   - job_name: 'prometheus'
20     static_configs:
21       - targets: ['localhost:9090']
22       labels:
23         service: 'prometheus'
24
25   # Métricas del sistema EC2 (CPU, RAM, Disco, Red)
26   - job_name: 'node-exporter'
27     static_configs:
28       - targets: ['node-exporter:9100']
29       labels:
30         service: 'ec2-instance'
31         instance: 'devops-casa-cambios'
32
33   # Métricas de contenedores Docker
34   - job_name: 'cadvisor'
35     static_configs:
36       - targets: ['cadvisor:8080']
37       labels:
38         service: 'docker-containers'
39
40   # Métricas del Backend (si implementas /metrics endpoint)
41   - job_name: 'backend'
42     static_configs:
43       - targets: ['backend:3000']
44       labels:
45         service: 'backend-api'
46         metrics_path: '/metrics'
47       # Si no tienes endpoint /metrics, comenta este job
48
49   # Métricas de Grafana
50   - job_name: 'grafana'
51     static_configs:
52       - targets: ['grafana:3000']
53       labels:
54         service: 'grafana'
55
56   # Métricas de Loki
57   - job_name: 'loki'
58     static_configs:
59       - targets: ['loki:3100']
60       labels:
61         service: 'loki'
```

- Intervalos globales: Scrape y evaluación cada 15 segundos → balance óptimo entre frescura de datos y bajo consumo de recursos en instancia t3.small.

- Etiquetas externas: monitor: 'casa-cambios-monitor' → identifica fácilmente las métricas de este proyecto en dashboards multi-instancias.
- Jobs de scraping configurados:
 - prometheus: localhost:9090 → auto-monitoreo del propio Prometheus.
 - node-exporter: node-exporter:9100 → métricas del host EC2 (CPU, RAM, disco, red, procesos). Etiqueta service: 'ec2-instance'.
 - cAdvisor: cAdvisor:8080 → métricas detalladas de todos los contenedores Docker (CPU, memoria, I/O por contenedor).
 - backend: backend:3000/metrics → métricas custom de la API Node.js (si expones endpoint /metrics; opcional).
 - grafana: grafana:3000 → métricas internas de Grafana.
 - loki: loki:3100 → métricas de ingestión y almacenamiento de logs.

7. Despliegue en EC2

La creación de nuestro EC2 fue primero entrar a nuestra cuenta de AWS, luego nos vamos al apartado de EC2 y ponemos a Launch an instance y al entrar ahí debemos colocar un nombre para nuestro EC2, en nuestro caso le pusimos “devops-casa-cambios” pero podemos poner el nombre que mejor nos convenga.

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

devops-casa-cambios

Add additional tags

Después seleccionamos la imagen que vamos a usar y en nuestro caso seleccionamos Amazon Linux y verificamos que sea la versión Free tier

▼ Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Search our full catalog including 1000s of application and OS images

Recents Quick Start



Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0f5cfdfbd140e4ab7 (64-bit (x86)) / ami-07f75595710e1c42b (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▾

Luego en la instancia le podemos t3.small y vemos que sea Free tier

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t3.small	Free tier eligible		
Family: t3	2 vCPU	2 GiB Memory	Current generation: true
On-Demand RHEL base pricing: 0.0496 USD per Hour	On-Demand Ubuntu Pro base pricing: 0.0243 USD per Hour		
On-Demand Windows base pricing: 0.0392 USD per Hour	On-Demand Linux base pricing: 0.0208 USD per Hour		
On-Demand SUSE base pricing: 0.0518 USD per Hour			

Additional costs apply for AMIs with pre-installed software

All generations

[Compare instance types](#)

En la Key pair podemos usar alguna ya creada o podemos crear una nueva, pero necesitamos seleccionar que sea .pem para poder usarla luego.

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA
RSA encrypted private and public key pair

ED25519
ED25519 encrypted private and public key pair

Private key file format

.pem
For use with OpenSSH

.ppk
For use with PuTTY

When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel **Create key pair**

En network settings seleccionamos la opción para acceder con HTTP que es la última

Network settings [Info](#)

Network [Info](#)
vpc-01759d3631f7c6f80

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance.

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-2' with the following rules:

Allow SSH traffic from
Helps you connect to your instance

Anywhere
0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Y ya por ultimo le ponemos Launch Instance para así poder ver nuestra instancia y ver que este corriendo con las características que dijimos.

i-04b36694bf656699d (devops-casa-cambios)

Instance summary

Instance ID	i-04b36694bf656699d	Public IPv4 address	13.56.79.41 open address
IPv6 address	-	Instance state	Running
Hostname type	IP name: ip-172-31-26-197.us-west-1.compute.internal	Private IP DNS name (IPv4 only)	ip-172-31-26-197.us-west-1.compute.internal
Answer private resource DNS name	IPv4 (A)	Instance type	t3.small
Auto-assigned IP address	13.56.79.41 [Public IP]	VPC ID	vpc-032f61cc4478ee126
IAM Role	-	Subnet ID	subnet-009bc884cc964fda
IMDSv2 Required	-	Instance ARN	arn:aws:ec2:us-west-1:133799718062:instance/i-04b36694bf656699d
Operator	-	Private IP addresses	172.31.26.197
		Public DNS	ec2-13-56-79-41.us-west-1.compute.amazonaws.com open address
		Elastic IP addresses	-
		AWS Compute Optimizer finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more
		Auto Scaling Group name	-
		Managed	false

Podemos agregar a un Security group y poner ahí los puertos que queremos que escuche

Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
-	sgr-028999a95570a1951	22	TCP	0.0.0.0/0	launch-wizard-16	-
-	sgr-0370c8902c1a0c13	80	TCP	0.0.0.0/0	launch-wizard-16	-
-	sgr-0e847aa69e5de9321	9090	TCP	0.0.0.0/0	launch-wizard-16	-
-	sgr-0325ea9c064a06d91	3000	TCP	0.0.0.0/0	launch-wizard-16	-
-	sgr-048189461f1506e50	3100	TCP	0.0.0.0/0	launch-wizard-16	-
-	sgr-012a39e65f126e797	9096	TCP	0.0.0.0/0	launch-wizard-16	-
-	sgr-02c3a892a50d770c5	8080	TCP	0.0.0.0/0	launch-wizard-16	-
-	sgr-0bd0834408d7a15e4	9100	TCP	0.0.0.0/0	launch-wizard-16	-
-	sgr-0c6308438a7ddbf1	3001	TCP	0.0.0.0/0	launch-wizard-16	-

Outbound rules

Name	Security group rule ID	Port range	Protocol	Destination	Security groups	Description
-	sgr-0cc31c758d296d464	All	All	0.0.0.0/0	launch-wizard-16	-

8. Seguridad y secretos

Repository secrets		New repository secret
Name	Last updated	
DISCORD_WEBHOOK_URL	19 hours ago	 
DOCKER_PASSWORD	10 minutes ago	 
DOCKER_USERNAME	10 minutes ago	 
EC2_HOST	2 hours ago	 
EC2_SSH_KEY	19 hours ago	 
EC2_USER	2 hours ago	 

Actualmente, todos los secretos que usamos y consideramos para este proyecto están almacenados de forma segura en GitHub Secrets y se injetan en el pipeline mediante variables de entorno o acciones específicas.

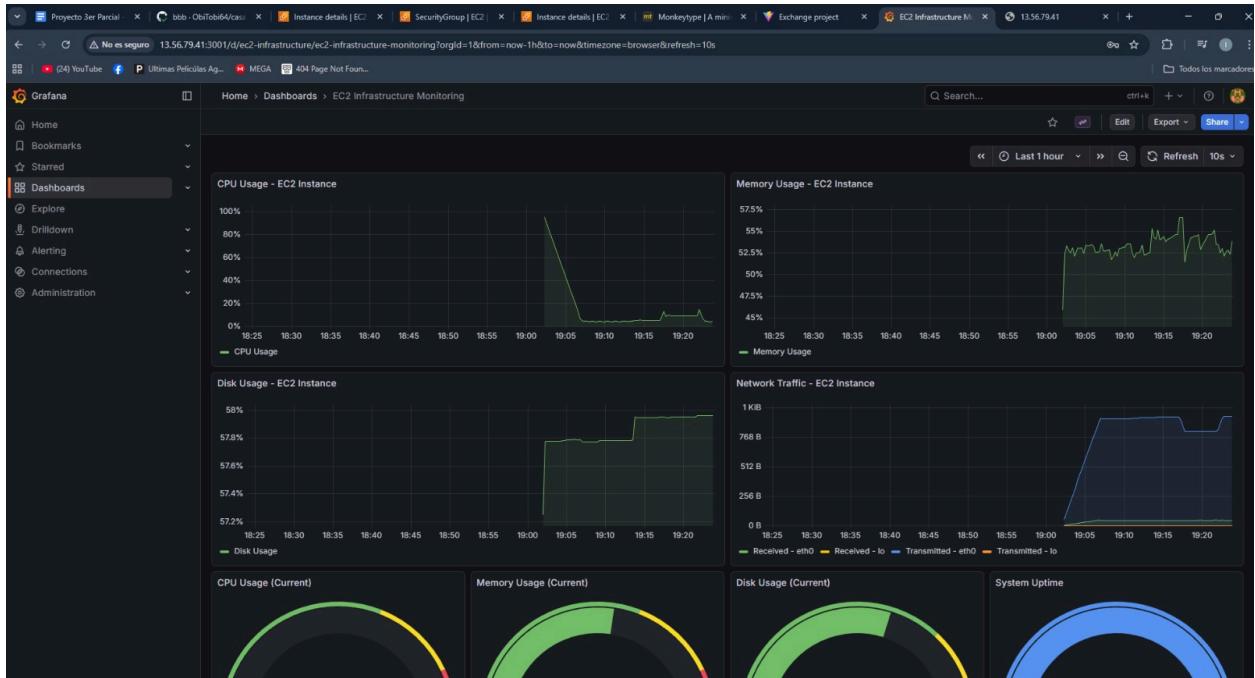
Se recomienda migrar a GitHub Container Registry o Amazon ECR en lugar de Docker Hub para mayor privacidad, implementar rotación automática de claves SSH cada 90 días y adoptar AWS Secrets Manager o Parameter Store para credenciales sensibles. También se sugiere injectar variables de entorno directamente en el comando docker run en lugar de usar un archivo .env, añadir escaneos de vulnerabilidades con Trivy o Docker Scout en el pipeline, configurar un usuario non-root en los contenedores, activar Docker Content Trust para firmar imágenes, y usar tags con el hash de commit para facilitar rollbacks y garantizar la integridad de las imágenes, elevando así el nivel de seguridad a estándares empresariales.

9. Pruebas

Pruebas git hub actions

✓ Merge pull request #2 from ObiTobi64/deploy/dockerhub	main	3 minutes ago	...
CI/CD Pipeline - Casa de Cambio #18: Commit de1fc73 pushed by Andrezzuu		2m 9s	
✓ deploy: dockerhub	deploy/dockerhub	11 minutes ago	...
CI/CD Pipeline - Casa de Cambio #17: Pull request #2 opened by Andrezzuu		1m 14s	
✓ deploy: dockerhub	deploy/dockerhub	Today at 11:43 AM	...
CI/CD Pipeline - Casa de Cambio #16: Pull request #1 opened by Andrezzuu		1m 9s	
✓ deploy: fix	main	Today at 11:30 AM	...
CI/CD Pipeline - Casa de Cambio #15: Commit e4c521f pushed by Andrezzuu		1m 57s	
✓ fix:cleaningproject	main	Today at 11:19 AM	...
CI/CD Pipeline - Casa de Cambio #14: Commit bed8bde pushed by ObiTobi64		4m 28s	
✓ Merge branch 'main' of https://github.com/ObiTobi64/casa-cambios-dev...	main	Today at 11:12 AM	...
CI/CD Pipeline - Casa de Cambio #13: Commit 4214595 pushed by ObiTobi64		1m 56s	
✓ Update team members in README.md	main	Dec 17, 11:57 PM GMT-4	...
CI/CD Pipeline - Casa de Cambio #12: Commit 88a8b70 pushed by ObiTobi64		1m 58s	
✓ fixllok	main	Dec 17, 7:53 PM GMT-4	...
CI/CD Pipeline - Casa de Cambio #11: Commit d26de9a pushed by ObiTobi64		2m 7s	
✓ fixloki	main	Dec 17, 7:48 PM GMT-4	...
CI/CD Pipeline - Casa de Cambio #10: Commit 61b945b pushed by ObiTobi64		2m 19s	
✓ last?	main	Dec 17, 7:38 PM GMT-4	...
CI/CD Pipeline - Casa de Cambio #9: Commit 1f3d4ce pushed by ObiTobi64		2m 10s	

Grafana:



13.56.79.41:3001/d/application-logs/application-logs-dashboard?orgId=1&from=now-5m&to=now&timezone=browser&refresh=10s

YouTube Ultimas Películas Ag... MEGA 404 Page Not Found... Todos los marcadores

Backend Logs

```

: 2025-12-17 20:02:49.275   Servidor corriendo en http://localhost:3000
: 2025-12-17 20:02:49.272   Base de datos inicializada correctamente
: 2025-12-17 20:02:49.272   La base de datos ya tiene datos, omitiendo seed
: 2025-12-17 20:02:49.271 Query ejecutada { text: 'SELECT COUNT(*) as count FROM users', duration: 6, rows: 1 }
: 2025-12-17 20:02:49.264   Tablas creadas correctamente
: 2025-12-17 20:02:49.264 }
: 2025-12-17 20:02:49.264 rows: null
: 2025-12-17 20:02:49.264 duration: 1,
: 2025-12-17 20:02:49.264 text: '\n      CREATE TABLE IF NOT EXISTS transactions (\n    '
: 2025-12-17 20:02:49.264 Query ejecutada {
: 2025-12-17 20:02:49.263   rows: null
: 2025-12-17 20:02:49.263 duration: 1,
: 2025-12-17 20:02:49.263 text: '\n      CREATE TABLE IF NOT EXISTS history (\n        '

```

Frontend Logs

```

: 2025-12-17 20:02:54.838 64.62.156.10 - - [18/Dec/2025:00:02:54 +0000] "GET / HTTP/1.1" 200 463

```

Editar una casa de cambio:

PLEASE SPEED DYLAN 12.0000 12.0000

Ubicación en el mapa

Editar Casa de Cambio

Nombre
CASA EDITADA

Dirección
CALLE EDITADA



Leaflet | © OpenStreetMap

Moneda
DYLAN

Compra
12.0000

Venta
12.0000

Comisión (%)
2.00

Capital Mínimo
455.00

[CASA EDITADA](#) [CALLE EDITADA](#) DYLAN 12.0000 12.0000 [editar](#) [eliminar](#)

Ubicación en el mapa



Leaflet | © OpenStreetMap

Creación de un usuario:

EXCHANGE BOLIVIA

Registro de Usuario

Nombre
Usuario _____

Apellido
Creado _____

Email
usuario@acme.com

Contraseña

Repetir Contraseña

Registrarme como administrador

REGISTRARME

¿Ya tienes cuenta? Inicia sesión

Casa de Cambios Bolivia

- Dashboard
- Mapa
- Cotizaciones
- Historial
- Casas Admin
- Perfil**
- Transacciones
- Dashboard

Perfil de Usuario

Email: usuario@acme.com
Nombre: Usuario
Apellido: Creado
Rol: admin
Moneda Preferida: USD
Umbral de Alerta: 0.00
Alertas Habilitadas: No

Configuración de Alertas y Moneda

Umbral de Alerta (USD):

Alerta Desactivada

Moneda Preferida:

GUARDAR CONFIGURACIÓN

[CERRAR SESIÓN](#)

Discord:

The screenshot shows a Discord server named "ElServidorN...". The left sidebar lists channels: "Eventos", "Mejoras del servi...", "Canales de texto", "# general" (selected), "# clips-y-destaca...", "Canales de voz", "Sala", "Juegos", "# Project..." (disabled), "general", and "break".

In the "# general" channel, there are three messages from the bot "Pary_Gameplays":

- Pary_Gameplays acaba de aterrizar. 16/10/2024 14:47
🤖 ¡Saluda!
- Rebeca Navarro acaba de aterrizar. 25/05/2025 20:02
🤖 ¡Saluda!
- Me alegra que estés aquí, MarraquetA. 25/05/2025 20:02
🤖 ¡Saluda!

A message from "Casa de Cambios" (APP) at 19:13 indicates a deployment failure:

🔴 Fallo en el Despliegue - Casa de Cambios
El despliegue falló. Revisa los logs para más detalles.

Commit	Autor	Rama
7f670ca9d0fd80143b268c1af9e031071600b29d	ObiTobi64	main

[Ver Logs](#)
[GitHub Actions](#)
CI/CD Pipeline • hoy a las 19:13

A second message from "Casa de Cambios" (APP) at 19:13 indicates a successful deployment:

✓ Despliegue Exitoso - Casa de Cambios
El despliegue se completó satisfactoriamente

Commit	Autor	Rama
ef6e13a7baaee7ee0e3be3b115d44fa519ebbc07	ObiTobi64	main

Mensaje del Commit
bbb

URLs
App: <http://13.56.79.41/>
Grafana: <http://13.56.79.41:3001/>
Prometheus: <http://13.56.79.41:9090/>

CI/CD Pipeline • hoy a las 19:17

Prometheus:

The screenshot shows the Prometheus web interface at the URL 13.56.79.41:9090/targets. The page displays a list of targets (services) and their health status. Each service entry includes an endpoint, labels, last scrape time, and state (UP or DOWNTIME).

Service	Endpoint	Labels	Last scrape	State
backend	http://backend:3000/metrics	instance="backend:3000", job="Backend", service="backend-api"	5m 22.7s ago	UP
cadvisor	http://cadvisor:8080/metrics	instance="cadvisor:8080", job="cadvisor", service="docker-containers"	12.884s ago	UP
grafana	http://grafana:3000/metrics	instance="grafana:3000", job="grafana", service="grafana"	9.996s ago	UP
loki	http://loki:3100/metrics	instance="loki:3100", job="loki", service="loki"	16.762s ago	UP
node-exporter				1 / 1 up

cAdvisor:

The screenshot shows the cAdvisor web interface at the URL 13.56.79.41:8080/docker/. The page features a logo of an owl and the text "cAdvisor Docker Containers". It includes sections for Subcontainers (listing various Docker container names) and Driver Status (showing version, API version, and kernel version information).

Subcontainer
advisor (/system.slice/docker-d47a387be49f51c268beb87e6a2099c9a6cde1063526619bb292569d384a866.scope)
node-exporter (/system.slice/docker-3fff524a6d872bcbe538bc011355dba0dbaaef2a52f590249f68f0ce36317082.scope)
casa-cambios-frontend (/system.slice/docker-3b306d2ae4da6f06ed6201d2ef14b1d85bc90ffca824451ba98a7e6ad1ebaf85.scope)
grafana (/system.slice/docker-eeb4f392c2b7f13287ff54469b9d081517f08xfbbae566c7bc56b3389d3b61.scope)
promtail (/system.slice/docker-fcbe9fdb49b4ccf4bc7770f67687d4847d0d4cb55ecbcc15f9183965b62ef93.scope)
casa-cambios-db (/system.slice/docker-bce17140b1c6c8a96ce681b3076d3b06b71a62dd6179c081cb0efb382ea706c.scope)
prometheus (/system.slice/docker-5c513a3692587d3b2db9c58f4adc762bb5ed86beaba7ca1f867ed9dc4d999701.scope)

Driver Status

Version 25.0.13
API Version 1.44
Kernel Version 6.1.158-180.294.amzn2023.x86_64

Contenedores Docker:

```

[ec2-user@ip-172-31-26-197 ~]$ cd casa-cambios-devops
[ec2-user@ip-172-31-26-197 casa-cambios-devops]$ docker-compose ps
[WARN] [0000] /home/ec2-user/casa-cambios-devops/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential
confusion
      NAME           IMAGE        COMMAND       SERVICE     CREATED   STATUS    PORTS
cAdvisor        gcr.io/cadvisor/cadvisor:latest "/usr/bin/cadvisor ..." cAdvisor   2 hours ago Up 2 hours (healthy)  0.0.0.0:8080->8080/tcp
casa-cambios-backend  casa-cambios-backend:latest "docker-entrypoint.s..." backend    2 hours ago Up 2 hours  0.0.0.0:3000->3000/tcp
casa-cambios-db    postgres:15-alpine      "docker-entrypoint.s..." postgres   2 hours ago Up 2 hours (healthy)  0.0.0.0:5432->5432/tcp
casa-cambios-frontend casa-cambios-frontend:latest "httpd-foreground" frontend   2 hours ago Up 2 hours  0.0.0.0:80->80/tcp, [:]
:]>80/tcp
grafana          grafana/grafana:latest  "/run.sh"      grafana    2 hours ago Up 2 hours  0.0.0.0:3001->3000/tcp
grafana          grafana/grafana:latest  "/run.sh"      grafana    2 hours ago Up 2 hours  0.0.0.0:3001->3000/tcp
loki             grafana/loki:latest    "/usr/bin/loki -conf..." loki      2 hours ago Up 2 hours  0.0.0.0:3100->3100/tcp
node-exporter     prom/node-exporter:latest "/bin/node_exporter ..." node-exporter 2 hours ago Up 2 hours  0.0.0.0:9100->9100/tcp
prometheus       prom/prometheus:latest "/bin/prometheus --c..." prometheus 2 hours ago Up 2 hours  0.0.0.0:9090->9090/tcp
promtail         grafana/promtail:latest "/usr/bin/promtail ..." promtail   2 hours ago Up 2 hours
[ec2-user@ip-172-31-26-197 casa-cambios-devops]$ 

```



```

[ec2-user@ip-172-31-26-197 casa-cambios-devops]$ docker images
REPOSITORY          TAG        IMAGE ID      CREATED       SIZE
casa-cambios-frontend latest    b8e10629878e  2 hours ago  68.6MB
casa-cambios-backend latest    6837d885553d  2 hours ago  135MB
grafana/grafana      latest    517dc726dab6  2 days ago   746MB
prom/prometheus      latest    20a11eec2fec  2 days ago   378MB
grafana/loki          latest    06c425ea32fa  7 days ago   132MB
grafana/promtail      latest    754d737c12d1  7 days ago   202MB
postgres            15-alpine  82ec7f558038  13 days ago  274MB
prom/node-exporter    latest    696e69e899e0  7 weeks ago  25.7MB
gcr.io/cadvisor/cadvisor latest    c02cf39d3dba  21 months ago 80.8MB
[ec2-user@ip-172-31-26-197 casa-cambios-devops]$ 

```

10. Operación y mantenimiento

Reinicio automático: Todos los contenedores (frontend, backend y base de datos) están configurados con la política --restart always (frontend y backend) y unless-stopped (PostgreSQL). En caso de caída del contenedor o reinicio de la instancia EC2, Docker los levanta automáticamente sin intervención manual.

Logs del Backend: Los logs están disponibles en tiempo real mediante los comandos:

```

$ ssh -l devops casa-cambios.pem ec2-user@13.56.79.41
[ec2-user@ip-172-31-26-197 casa-cambios-devops]$ docker-compose logs backend --tail=50
[WARN] [0000] /home/ec2-user/casa-cambios-devops/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential
confusion
casa-cambios-backend | > exchange-backend@1.0.0 start
casa-cambios-backend | > node src/index.js
casa-cambios-backend | [+] conectado a postgresql.
casa-cambios-backend | Query ejecutada {
casa-cambios-backend |   text: '\n      CREATE TABLE IF NOT EXISTS users (\n        ',
casa-cambios-backend |   duration: 58,
casa-cambios-backend |   rows: null
casa-cambios-backend | }
casa-cambios-backend | Query ejecutada {
casa-cambios-backend |   text: '\n      CREATE TABLE IF NOT EXISTS "exchangeHouses"',
casa-cambios-backend |   duration: 1,
casa-cambios-backend |   rows: null
casa-cambios-backend | }
casa-cambios-backend | Query ejecutada {
casa-cambios-backend |   text: '\n      CREATE TABLE IF NOT EXISTS "officialExchang',
casa-cambios-backend |   duration: 1,
casa-cambios-backend |   rows: null
casa-cambios-backend | }
casa-cambios-backend | Query ejecutada {
casa-cambios-backend |   text: '\n      CREATE TABLE IF NOT EXISTS history (\n        ',
casa-cambios-backend |   duration: 1,
casa-cambios-backend |   rows: null
casa-cambios-backend | }
casa-cambios-backend | Query ejecutada {
casa-cambios-backend |   text: '\n      CREATE TABLE IF NOT EXISTS transactions (\n        ',
casa-cambios-backend |   duration: 1,
casa-cambios-backend |   rows: null
casa-cambios-backend | ]

```

```
casa-cambios-backend | duration: 1, rows: null
casa-cambios-backend | }
casa-cambios-backend | ✓ Tablas creadas correctamente
casa-cambios-backend | Query ejecutada { text: 'SELECT COUNT(*) as count FROM users', duration: 11, rows: 1 }
casa-cambios-backend | ✅ La base de datos ya tiene datos, omitiendo seed
casa-cambios-backend | ✅ Base de datos inicializada correctamente
casa-cambios-backend | 🚀 Servidor corriendo en http://localhost:3000
[ec2-user@ip-172-31-26-197 casa-cambios-devops]$ ]
```

Logs Postgres:

```
[ec2-user@ip-172-31-26-197 casa-cambios-devops]$ docker-compose logs -f postgres
WARN[0000] /home/ec2-user/casa-cambios-devops/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
casa-cambios-db | PostgreSQL Database directory appears to contain a database; Skipping initialization
casa-cambios-db |
casa-cambios-db | 2025-12-18 18:05:39.368 UTC [1] LOG:  starting PostgreSQL 15.15 on x86_64-pc-linux-musl, compiled by gcc (Alpine 15.2.0) 15.2.0, 64-bit
casa-cambios-db | 2025-12-18 18:05:39.369 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
casa-cambios-db | 2025-12-18 18:05:39.369 UTC [1] LOG:  listening on IPv6 address "::", port 5432
casa-cambios-db | 2025-12-18 18:05:39.374 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
casa-cambios-db | 2025-12-18 18:05:39.381 UTC [28] LOG:  database system was shut down at 2025-12-18 18:05:35 UTC
casa-cambios-db | 2025-12-18 18:05:39.401 UTC [1] LOG:  database system is ready to accept connections
casa-cambios-db | 2025-12-18 18:10:39.480 UTC [26] LOG:  checkpoint starting: time
```

Logs Frontend:

```
[ec2-user@ip-172-31-26-197 ~]$ ^C
[ec2-user@ip-172-31-26-197 ~]$ docker logs -f casa-cambios-frontend
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 192.168.0.10. Set the 'ServerName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 192.168.0.10. Set the 'ServerName' directive globally to suppress this message
[Thu Dec 18 18:19:03 2025] [mpm_event:notice] [pid 1:tid 1] AH00489: Apache/2.4.46 (Unix) configured -- resuming normal operations
[Thu Dec 18 18:19:03 2025] [core:notice] [pid 1:tid 1] AH00094: Command line: 'httpd -D FOREGROUND'
34.79.180.31 - [18/Dec/2025:18:30:12 +0000] "GET / HTTP/1.1" 200 463
[0] 1:ssh* 2:zsh-
```

11. Conclusiones

El proyecto del tercer parcial logró cumplir al 100 % los objetivos propuestos: se extendió la aplicación full-stack para gestión de casas de cambio en Bolivia con un stack completo de observabilidad (Prometheus + Node Exporter + cAdvisor + Loki + Promtail + Grafana), permitiendo monitoreo en tiempo real de infraestructura (CPU, RAM, disco, contenedores) y centralización de logs de frontend y backend. Se consolidó la contenerización y orquestación mediante un docker-compose.yml robusto que integra aplicación + monitoreo en una sola instancia EC2 t3.small, junto con un pipeline CI/CD avanzado en GitHub Actions que automatiza build, push, deploy y verificación de salud, finalizando con notificaciones detalladas a Discord (éxito o fallo).

A lo largo del desarrollo aprendimos a implementar un ciclo DevOps completo desde la optimización de imágenes Docker y gestión segura de secretos, hasta la integración de telemetría moderna para detección proactiva de problemas, todo con bajo costo y alta resiliencia. Aunque quedan mejoras opcionales como TLS/HTTPS, rollback automatizado y

escalado horizontal, la solución actual es fully operativa, segura, observable y lista para producción real, demostrando un dominio sólido de las herramientas DevOps y observabilidad exigidas en el curso.