

T H A R S E O

PROJETO INTEGRADOR

Alberto Ribeiro da Silva
Alessandra Sanches
Caio Ribeiro
Flavio Domingues Vieira
Richard Guedes Ribeiro da Silva

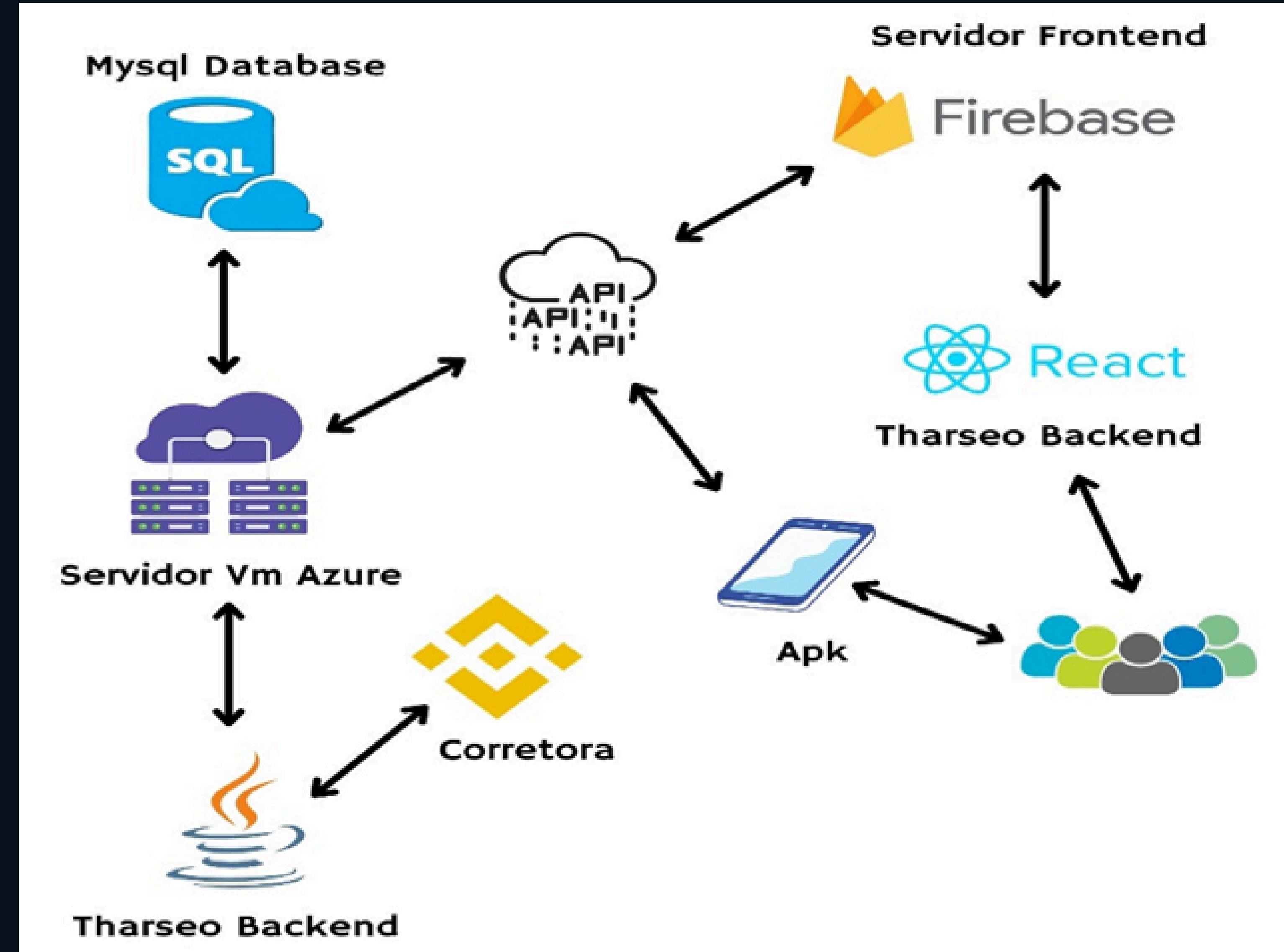
THARSEO

Tharseo Cripto é um SaaS que tem como objetivo automatizar e gerenciar negociações de forma automática 7 dias por semana e 24 horas por dia de acordo com a preferência de configuração do usuário.

Ele funciona de forma semelhante a um plugin, irá fazer o intermédio entre o usuário e a corretora. Ele funciona de forma automática, então conforme o usuário configurar, executará as ordens e realizara profits com seus alvos definidos.

INFRAESTRUTURA ECOSISTEMA THARSEO

T H A R S E O

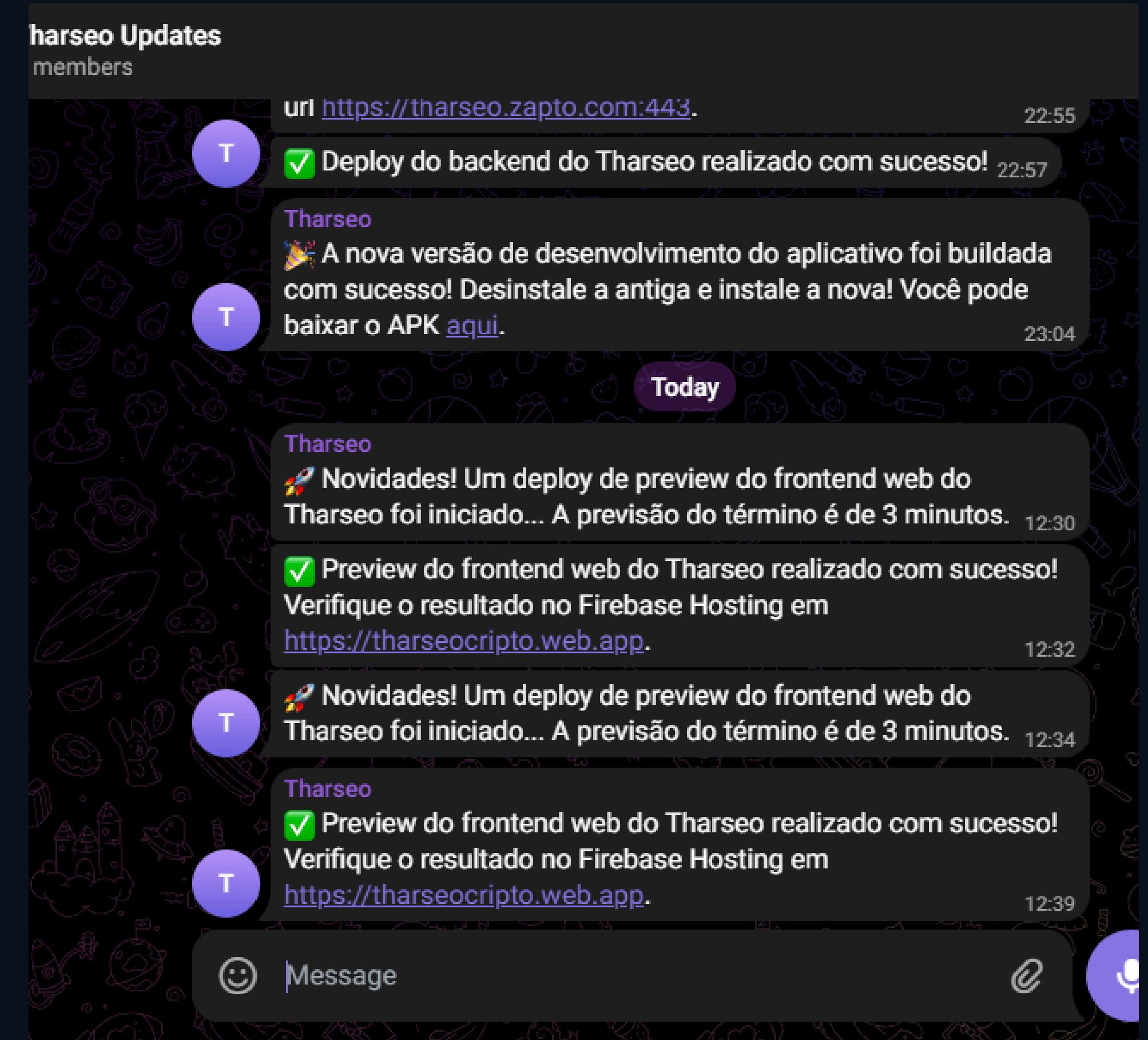


INFRAESTRUTURA ECOSISTEMA THARSEO - NOTIFICAÇÕES

R S E O

```
# Step 13: Notify on success with link to APK
- name: Notify on success with link to APK
  if: success()
  run: |
    ARTIFACT_URL="http://tharseo.zapto.org:8500/app-release.apk"
    MESSAGE="👉 A nova versão de desenvolvimento do aplicativo foi buildada com sucesso! Desinstale a antiga e instale a nova! Você pode baixar o APK aqui."
    curl -s -X POST https://api.telegram.org/bot${{ secrets.TELEGRAM_BOT_TOKEN }} \
      -d chat_id=${{ secrets.TELEGRAM_CHAT_ID }} \
      -d text="${MESSAGE}" \
      -d parse_mode="Markdown"

# Step 14: Notify on failure
- name: Notify on failure
  if: failure()
  run: |
    MESSAGE="❗️ Ocorreu um erro durante o processo de build do projeto."
    curl -s -X POST https://api.telegram.org/bot${{ secrets.TELEGRAM_BOT_TOKEN }} \
      -d chat_id=${{ secrets.TELEGRAM_CHAT_ID }} \
      -d text="${MESSAGE}" \
      -d parse_mode="Markdown"
```





SERVIÇOS AZURE

05

^ Fundamentos

Grupo de recursos ([mover](#)) : [tharseo](#)

Status : Em execução

Local : East US

Assinatura ([mover](#)) : [Azure for Students](#)

ID da Assinatura : 2c396bf6-96da-46df-884b-084b54a6e4e1

Sistema operacional : Linux (ubuntu 22.04)

Tamanho : Standard B1s (1 vcpu, 1 GiB de memória)

Endereço IP público : [52.168.150.124](#)

Rede virtual/sub-rede : [backendtharseo-vnet/default](#)

Nome DNS : [Não configurado](#)

Estado de integridade : -

Horário criado : 06/09/2024, 00:22 UTC

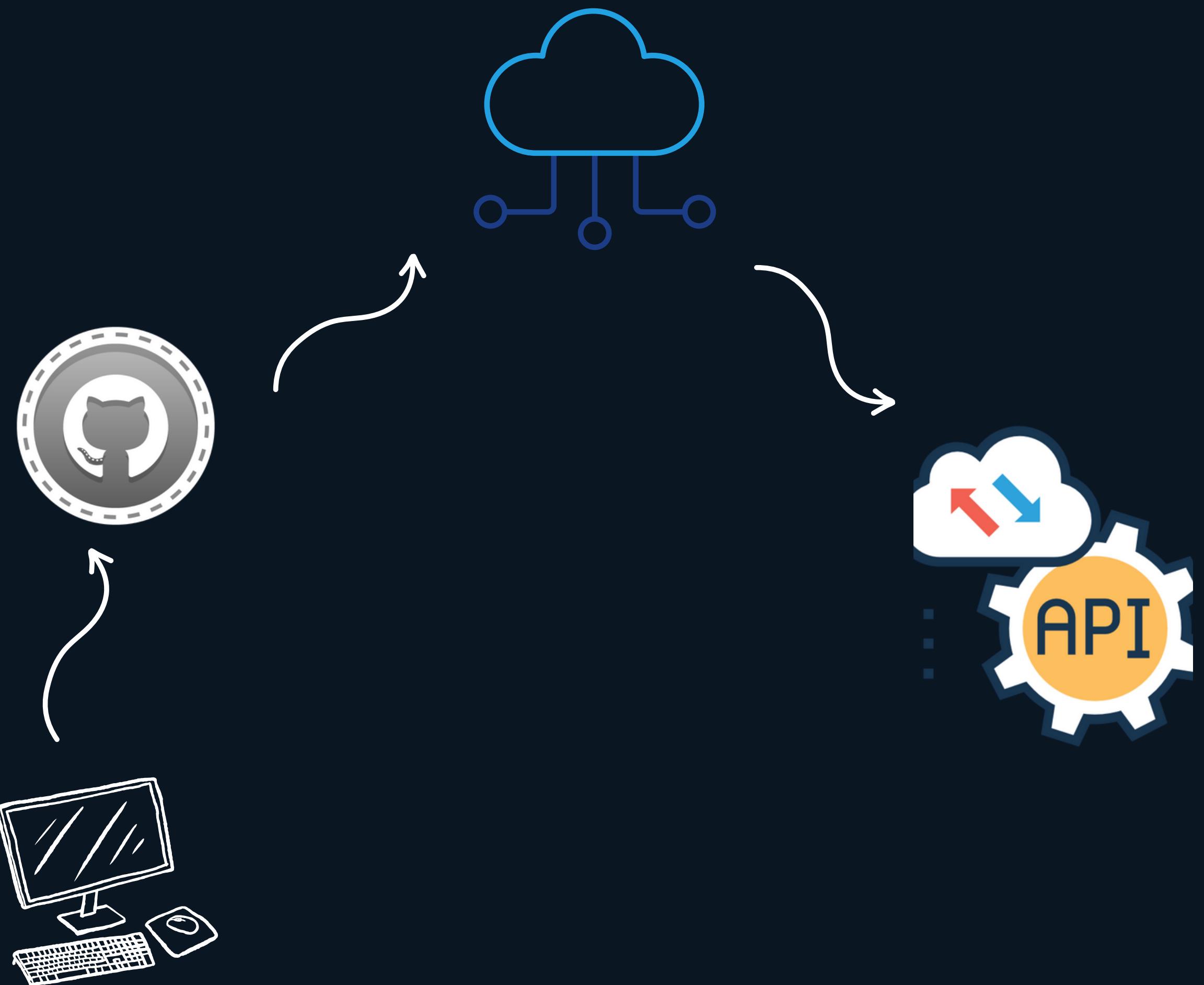
| BAIXO CUSTO COMPARADO A KUBERNETES

| FACILIDADE DE INTEGRAÇÃO COM OUTROS MICROSERVIÇOS

| ALTA DISPONIBILIDADE

INFRA BACKEND - CONTEXTO

T H A R S E O



07

INFRA BACKEND - DEPLOY COM GIT ACTIONS

```
name: Deploy Backend to Container in AzureVM

on:
  push:
    branches:
      - main

jobs:
  tests-backend:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v1

      - name: Set up Java
        uses: actions/setup-java@v2
        with:
          java-version: '17'
          distribution: 'temurin'

      - name: Run Tharseo Tests
        run: mvn clean test

      continue-on-error: false

  build-backend:
    needs:
      - tests-backend
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v2

  jobs:
    build-backend:
      steps:
        uses: actions/checkout@v2

        - name: Set up Docker Buildx
          uses: docker/setup-buildx-action@v1

        - name: Set up Java
          uses: actions/setup-java@v2
          with:
            java-version: '17'
            distribution: 'temurin'

        - name: Inject secrets into application.properties
          run:
            echo "spring.datasource.url=${{ secrets.DB_URL }}" >> src/main/resources/application.properties
            echo "spring.datasource.password=${{ secrets.DB_PASSWORD }}" >> src/main/resources/application.properties
            echo "spring.datasource.username=${{ secrets.DB_USERNAME }}" >> src/main/resources/application.properties

        - name: Build Tharseo
          run: mvn clean package
          Guedes, 05/09/2024 22:08 • Build(Repository) : Change repository and separate backend from frontend

        - name: Build Docker image
          run:
            echo ${{ secrets.DOCKER_PASSWORD }} | docker login -u ${{ secrets.DOCKER_USERNAME }} --password-stdin
            docker build -t richardguedesrib/tharseo-app:latest .

        - name: Push Docker image to Docker Hub
          run:
            docker push richardguedesrib/tharseo-app:latest

  deploy-backend:
    needs:
      - tests-backend
      - build-backend
```

- Files
- Tool
- Acti
- Sett





AUTOMATIZAÇÃO COM DIVERSOS PLUGINS FÁCIL GERENCIAMENTO DE INFORMAÇÕES SENSÍVEIS

| Name | Last updated | |
|---|--------------|--|
| DB_PASSWORD | 1 hour ago | |
| DB_URL | 1 hour ago | |
| DB_USERNAME | 1 hour ago | |
| DOCKER_PASSWORD | 4 months ago | |
| DOCKER_USERNAME | 4 months ago | |
| FIREBASE_SERVICE_ACCOUNT_THARSEOAUTOMATOR | 4 months ago | |
| FIREBASE_SERVICE_ACCOUNT_THARSEOFRONTEND | 4 months ago | |
| SONAR_TOKEN | 4 months ago | |
| SSH_PRIVATE_KEY | 4 months ago | |

INFRA BACKEND - MONITORAMENTO DE CONTAINER COM PORTAINER

ACESSO AO AMBIENTE DO CONTAINER

ACESSO AOS LOGS

GERENCIAMENTO DOS CONTAINERS

ACESSO VIA PORTA 9000 (Configurada na Azure)

The screenshot shows the Portainer.io interface with the following details:

- Header:** portainer.io COMMUNITY EDITION, tharseo
- Left Sidebar:** Home, local, Dashboard, Templates, Stacks, **Containers** (selected), Images.
- Top Bar:** Containers, Search, Start, Stop, Kill, Restart, Pause, Resume, Remove, Add container.
- Table Headers:** Name, State, Quick Actions, Stack, Image, Created, IP Address, Published Po.
- Table Data:**
 - friendly_payne: running, richardguedesrib/tharseo-app:latest, 2024-09-15 20:03:23, 172.17.0.2, 8080:8080
 - portainer: running, portainer/portainer-ce, 2024-09-15 20:53:44, 172.17.0.3, 9000:9000
- Bottom:** 1 item(s) selected, Items per page: 10.

INFRA BACKEND - SEGURANÇA - HTTPS/TLS

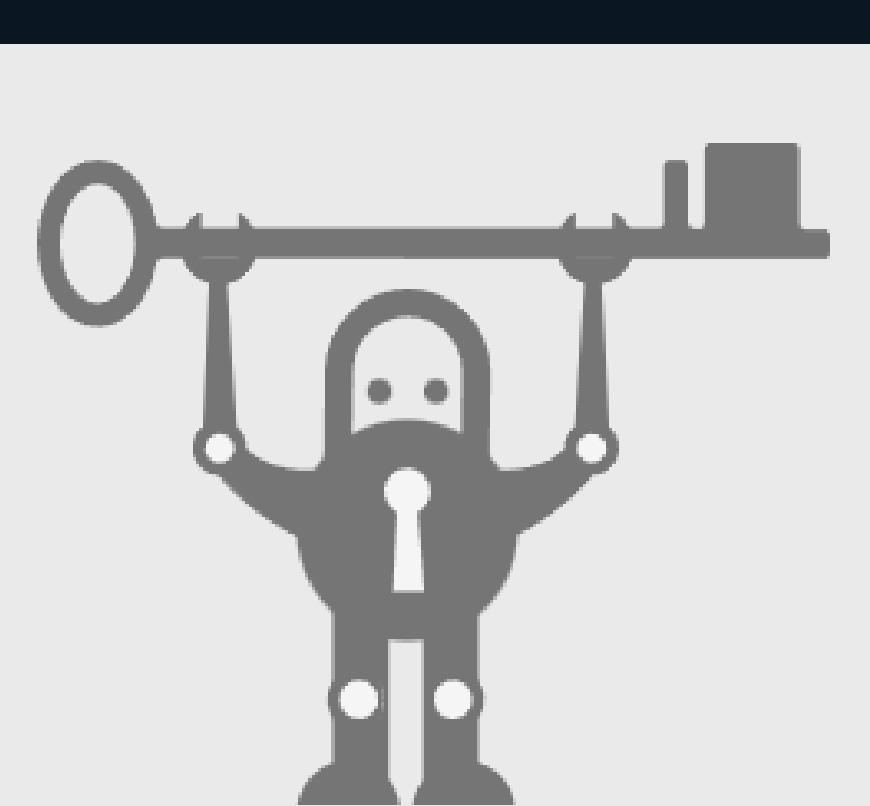
RECURSO HTTPS/TLS PARA SEGURANÇA ENTRE FRONT E BACK

Apontamento de Backend em domínio

| Apontamento de Backend em domínio | | | | |
|-----------------------------------|-----------------------------|----------------|------|---|
| Criar hostname | Pesquisar... | x | Q | |
| Nome de Host | Last Update | IP / Alvo | Type | DDNS Key |
| tharseo.zapto.org Active | Oct 25, 2024 15:08 PDT ! | 52.168.150.124 | A | Create DDNS Key Modificar x |

Or, run Certbot once to automatically get free HTTPS certificates forever.

[Get Certbot instructions](#)



Certificado Digital pela Certbot

SERVIDOR MYSQL AZURE

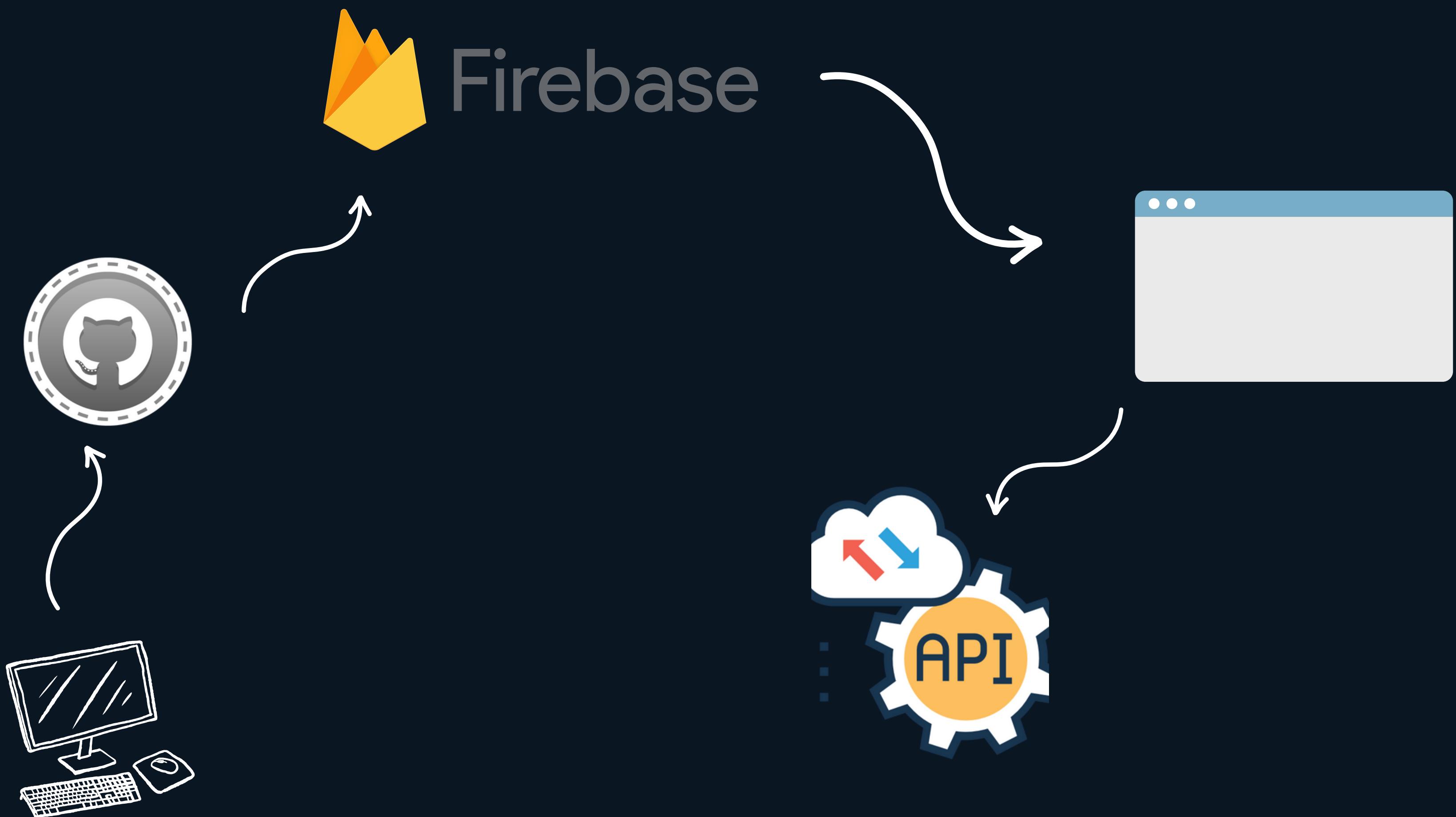
T H A R S E O

| Fundamentos | | Exibição JSON |
|---|---|---------------|
| Assinatura (mover) | : Azure for Students | |
| ID da Assinatura | : 2c396bf6-96da-46df-884b-084b54a6e4e1 | |
| Grupo de recursos (mover) | : tharseo | |
| Status | : Disponível | |
| Localização | : West US | |
| Marcações (editar) | : Adicionar marcas | |
| Nome do servidor | : tharseodatabase.mysql.database.azure.com | |
| Nome de logon do admi... | : database | |
| Configuração | : Intermitente, B1ms, 1 vCores, 2 GiB RAM, 20 GiB st... | |
| Versão MySQL | : 8.0 | |
| Zona de disponibilidade | : -- | |
| Criado em | : 2024-09-15 19:35:28.5411494 UTC | |

- | ALTA PERFORMANCE
- | FACILIDADE DE MIGRAÇÃO INTERNA
- | CUSTO MODERADO EM RELAÇÃO AO POSTGRES
- | POSSIBILIDADE DO USO DE REDE INTERNA AZURE

INFRA FRONTEND WEB - CONTEXTO

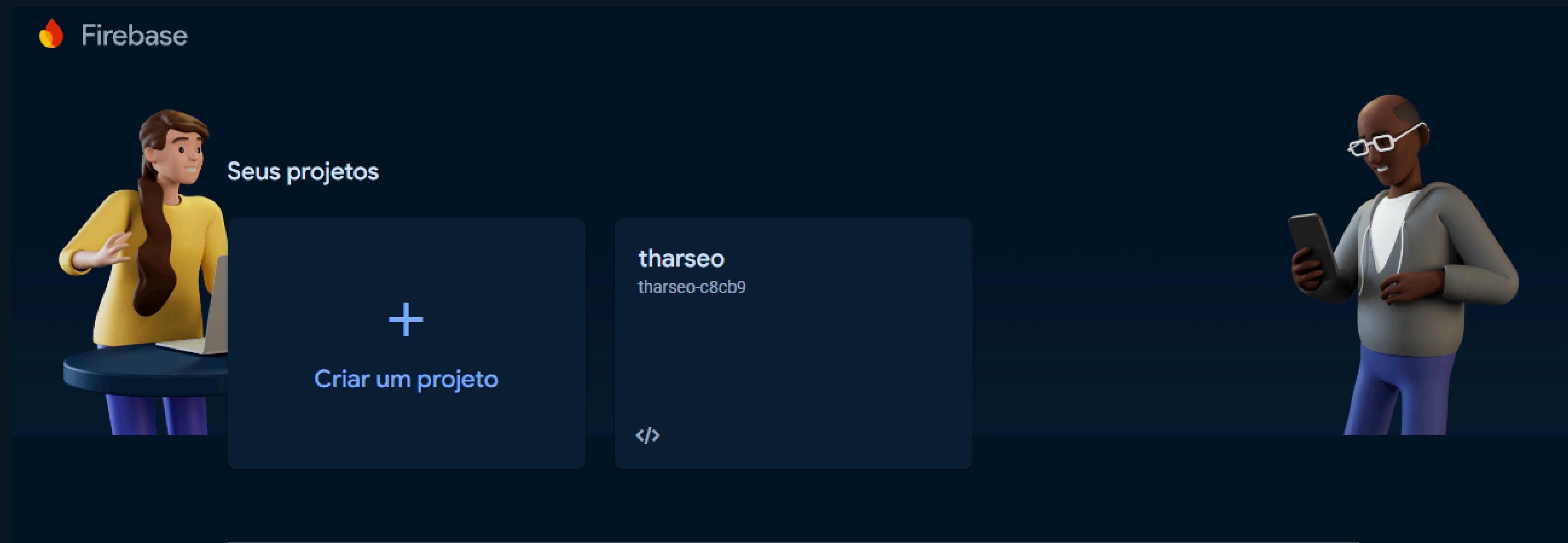
T H A R S E O



INFRA FRONTEND WEB - CLOUD FIREBASE

F A R S E O

- | CAMADA DE SERVIÇOS GRATUITA
- | FÁCIL INTEGRAÇÃO COM SERVIÇOS GOOGLE
- | FIREBASE CLI PARA FÁCIL CD/CI



INFRA FRONTEND WEB - DEPLOY COM GIT ACTIONS

```
5
6  on: pull_request
7
8    permissions:
9      checks: write
10     contents: read
11    pull-requests: write
12
13  jobs:
14    notify-start:
15      runs-on: ubuntu-latest
16      steps:
17        - name: Notify Telegram - Workflow Started
18          run: |
19            MESSAGE="⚡ Novidades! Um deploy de preview do frontend web do Tharseo foi iniciado para o PR #${{ github.event.number }}"
20            curl -s -X POST https://api.telegram.org/bot${{ secrets.TELEGRAM_BOT_TOKEN }}/sendMessage \
21              -d chat_id=${{ secrets.TELEGRAM_CHAT_ID }} \
22              -d text="${MESSAGE}" \
23              -d parse_mode="Markdown"
24
25  build_and_preview:
26    if: ${{ github.event.pull_request.head.repo.full_name == github.repository }}
27    needs: notify-start
28    runs-on: ubuntu-latest
29    steps:
30      - uses: actions/checkout@v4
31      - run: npm ci --legacy-peer-deps && npm run build --legacy-peer-deps
32      - uses: FirebaseExtended/action-hosting-deploy@v0
33        with:
34          repoToken: ${{ secrets.GITHUB_TOKEN }}
35          firebaseServiceAccount: ${{ secrets.FIREBASE_SERVICE_ACCOUNT_THARSEO_C8C89 }}
36          projectId: tharseo-c8cb9
37
38      - name: ZAP Scan
39        uses: zaproxy/action-full-scan@v0.10.0
40        with:
41          token: ${{ secrets.GITHUB_TOKEN }}
42          docker_name: 'ghcr.io/zaproxy/zaproxy:stable'
43          target: 'https://tharseocripto.web.app/'
44          rules_file_name: '.zap/rules.tsv'
45          cmd_options: '-a'
46          allow_issue_writing: false
47          artifact_name: 'zap_scan_report'
48
49      - name: Notify Telegram - Deploy Success
50        if: success()
51        run: |
52          MESSAGE="✅ Preview do frontend do Tharseo para o PR #${{ github.event.number }} realizado com sucesso! Verifique o r
53          curl -s -X POST https://api.telegram.org/bot${{ secrets.TELEGRAM_BOT_TOKEN }}/sendMessage \\"
```

INFRA FRONTEND WEB - TESTE DE SEGURANÇA A CADA DEPLOY

```
- name: ZAP Scan
  uses: zaproxy/action-full-scan@v0.10.0
  with:
    token: ${secrets.GITHUB_TOKEN}
    docker_name: 'ghcr.io/zaproxy/zaproxy:stable'
    target: 'https://tharseocripto.web.app/'
    rules_file_name: '.zap/rules.tsv'
    cmd_options: '-a'
    allow_issue_writing: false
    artifact_name: 'zap_scan_report'
```

 **ZAP by Checkmarx**

Site: <https://tharseocripto.web.app>

Generated on Sun, 10 Nov 2024 15:39:09

ZAP Version: 2.15.0

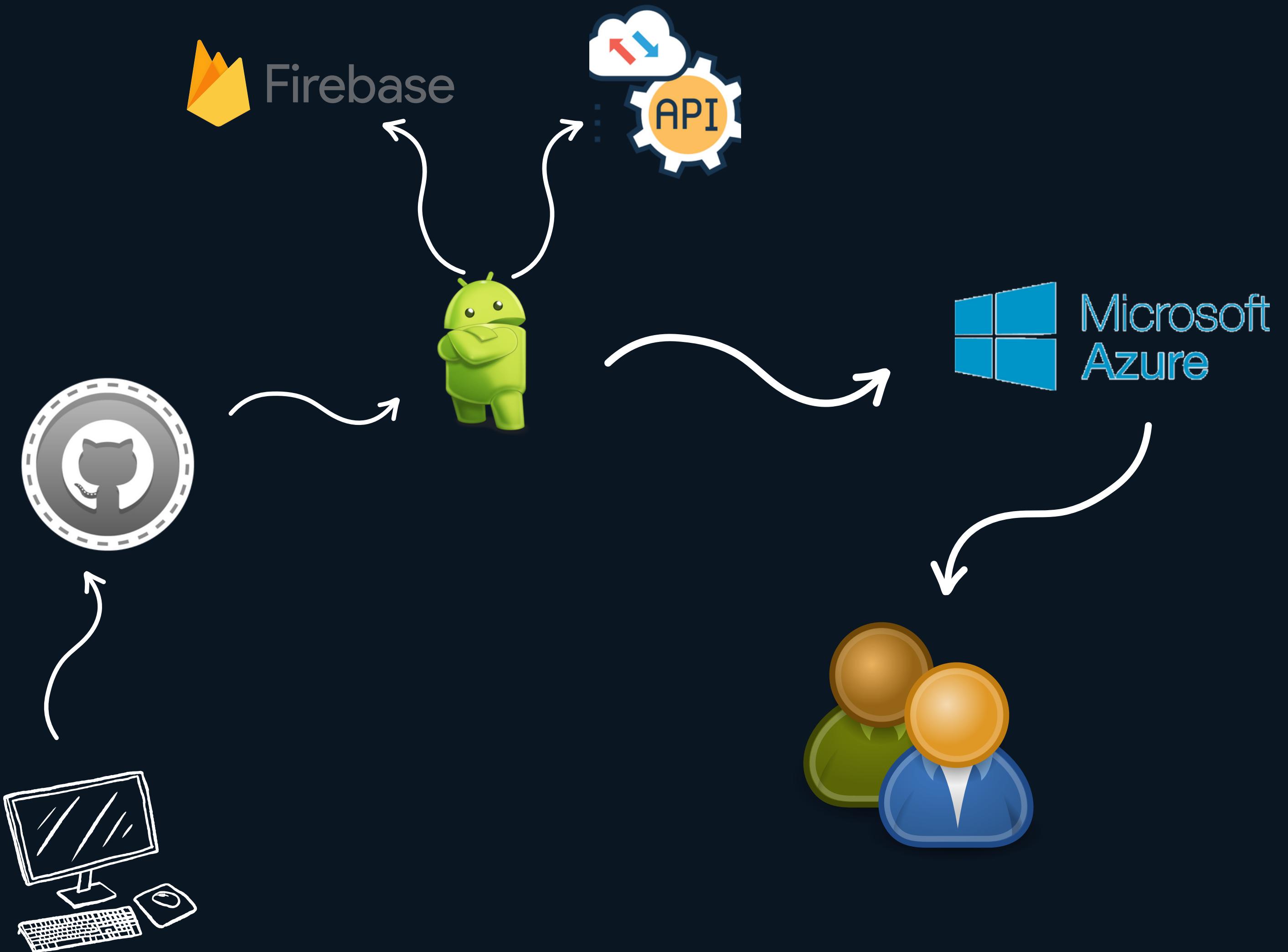
ZAP by Checkmarx

Summary of Alerts

| Risk Level | Number of Alerts |
|------------------|------------------|
| High | 0 |
| Medium | 6 |
| Low | 2 |
| Informational | 10 |
| False Positives: | 0 |

INFRA FRONTEND MOBILE - CONTEXTO

T H A R S E O



INFRA FRONTEND MOBILE- DEPLOY COM GIT ACTIONS

```
.github > workflows > | build-native.yml
  1   name: Build and Upload APK
  2
  3   on:
  4     push:
  5       branches:
  6         - develop-native
  7
  8   jobs:
  9     build:
 10       runs-on: ubuntu-latest
 11
 12     steps:
 13       # Step 1: Checkout repository
 14       - name: Checkout repository
 15         uses: actions/checkout@v2
 16
 17       # Step 2: Notify that the deploy has started
 18       - name: Notify that deploy has started
 19         run: |
 20           MESSAGE="💡 Deploy MOBILE na branch *develop-native* foi iniciado! A previsão para conclusão é de 12 minutos."
 21           curl -s -X POST https://api.telegram.org/bot${{ secrets.TELEGRAM_BOT_TOKEN }}/sendMessage \
 22             -d chat_id=${{ secrets.TELEGRAM_CHAT_ID }} \
 23             -d text="${MESSAGE}" \
 24             -d parse_mode="Markdown"
 25
 26       # Step 3: Set up Node.js
 27       - name: Set up Node.js
 28         uses: actions/setup-node@v2
 29         with:
 30           node-version: '18'
 31
 32       # Step 4: Install Java 17
 33       - name: Set up JDK 17
 34         uses: actions/setup-java@v2
 35         with:
 36           java-version: '17'
 37           distribution: 'temurin'
 38
 39       # Step 5: Install dependencies
 40       - name: Install dependencies
 41         run: npm install
 42
 43       # Step 6: Install React Native CLI
 44       - name: Install React Native CLI
 45         run: npm install -g react-native-cli
 46
 47       # Step 7: Clean Gradle Cache and Build the Android app
 48       - name: Build Android App (Release)
 49         ----
```

```
# Step 8: Upload the APK as an artifact
- name: Upload APK
  id: upload-apk
  uses: actions/upload-artifact@v3
  with:
    name: app-release.apk
    path: android/app/build/outputs/apk/release/app-release.apk

# Step 9: Download APK artifact
- name: Download APK artifact
  uses: actions/download-artifact@v3
  with:
    name: app-release.apk

# Step 10: Upload APK to Azure VM
- name: Upload APK to Azure VM
  uses: appleboy/scp-action@master
  with:
    host: ${{ secrets.THARSEO_BACKEND_HOST }}
    username: ${{ secrets.THARSEO_BACKEND_HOST_USER }}
    password: ${{ secrets.THARSEO_BACKEND_PASS }}
    port: 22
    source: app-release.apk
    target: /home/tharseo/apk/

# Step 11: Start Simple HTTP Server on Azure VM
- name: Start Simple HTTP Server on Azure VM
  uses: appleboy/ssh-action@master
  with:
    host: ${{ secrets.THARSEO_BACKEND_HOST }}
    username: ${{ secrets.THARSEO_BACKEND_HOST_USER }}
    password: ${{ secrets.THARSEO_BACKEND_PASS }}
    port: 22
    script: |
      # Verifica se o servidor já está rodando na porta 8500 e para, se necessário
      if lsof -i :8500; then
        echo "Parando o servidor HTTP antigo."
        fuser -k 8500/tcp
      fi

      # Inicia um novo servidor HTTP na porta 8500
      echo "Iniciando novo servidor HTTP."
      cd /home/tharseo/apk
```

INFRA FRONTEND MOBILE- DEPLOY COM GIT ACTIONS

```
username: ${ secrets.THARSEO_BACKEND_HOST_USER }
password: ${ secrets.THARSEO_BACKEND_PASS }
port: 22
script: |
  # Verifica se o servidor já está rodando na porta 8500 e para, se necessário
  if lsof -i :8500; then
    echo "Parando o servidor HTTP antigo."
    fuser -k 8500/tcp
  fi

  # Inicia um novo servidor HTTP na porta 8500
  echo "Iniciando novo servidor HTTP."
  cd /home/tharseo/apk
  nohup python3 -m http.server 8500 > /home/tharseo/apk/server.log 2>&1 &
  echo "Servidor HTTP iniciado com sucesso na porta 8500."

# Step 12: List download link
- name: List download link
  run: |
    echo "O APK está disponível para download em: http://52.168.150.124:8500/app-release.apk"

# Step 13: Notify on success with link to APK
- name: Notify on success with link to APK
  if: success()
  run: |
    ARTIFACT_URL="http://tharseo.zapto.org:8500/app-release.apk"
    MESSAGE="⚠ A nova versão de desenvolvimento do aplicativo foi buildada com sucesso! Desinstale a antiga e instale a nova! Você pode baixar o APK [aqui](${ARTIFACT_URL})"
    curl -s -X POST https://api.telegram.org/bot\${secrets.TELEGRAM\_BOT\_TOKEN}/sendMessage \
      -d chat_id=${secrets.TELEGRAM_CHAT_ID} \
      -d text="${MESSAGE}" \
      -d parse_mode="Markdown"

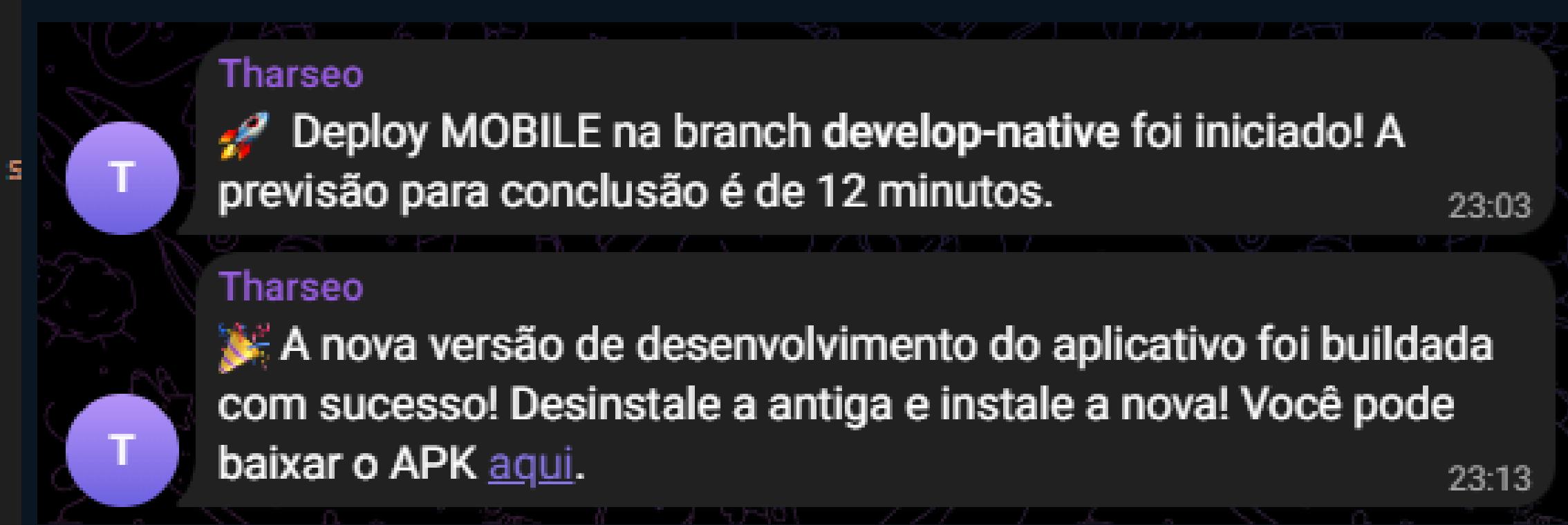
# Step 14: Notify on failure
- name: Notify on failure
  if: failure()
  run: |
    MESSAGE="❗ Ocorreu um erro durante o processo de build do aplicativo! Por favor, verifique os logs para mais informações."
    curl -s -X POST https://api.telegram.org/bot\${secrets.TELEGRAM\_BOT\_TOKEN}/sendMessage \
      -d chat_id=${secrets.TELEGRAM_CHAT_ID} \
      -d text="${MESSAGE}" \
      -d parse_mode="Markdown"
```

INFRA FRONTEND MOBILE - MICROSERVIÇO DE DRIVE

```
# Step 11: Start Simple HTTP Server on Azure VM
- name: Start Simple HTTP Server on Azure VM
  uses: appleboy/ssh-action@master
  with:
    host: ${{ secrets.THARSEO_BACKEND_HOST }}
    username: ${{ secrets.THARSEO_BACKEND_HOST_USER }}
    password: ${{ secrets.THARSEO_BACKEND_PASS }}
    port: 22
    script: |
      # Verifica se o servidor já está rodando na porta 8580 e para, se estiver
      if lsof -i :8580; then
        echo "Parando o servidor HTTP antigo."
        fuser -k 8580/tcp
      fi

      # Inicia um novo servidor HTTP na porta 8580
      echo "Iniciando novo servidor HTTP."
      cd /home/tharseo/apk
      nohup python3 -m http.server 8580 > /home/tharseo/apk/server.log
      echo "Servidor HTTP iniciado com sucesso na porta 8580."

# Step 12: List download link
- name: List download link
  run: |
    echo "O APK está disponível para download em: http://52.168.150.12"
```



INFRA FRONTEND MOBILE - APP

T H A R S E O

