G5 Securitas

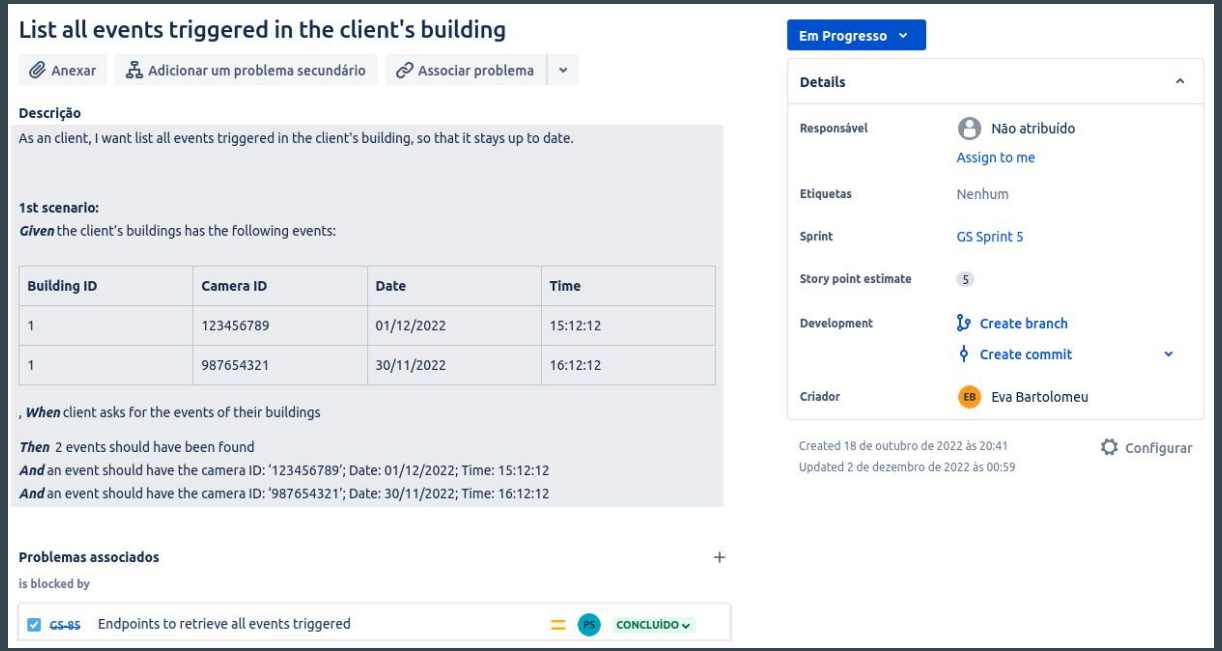
Course: 41492 - Software Engineering

Team: Diogo Vicente
Eduardo Santos
Eva Bartolomeu
João Ferreira
Pedro Sobral
Pedro Almeida

Index

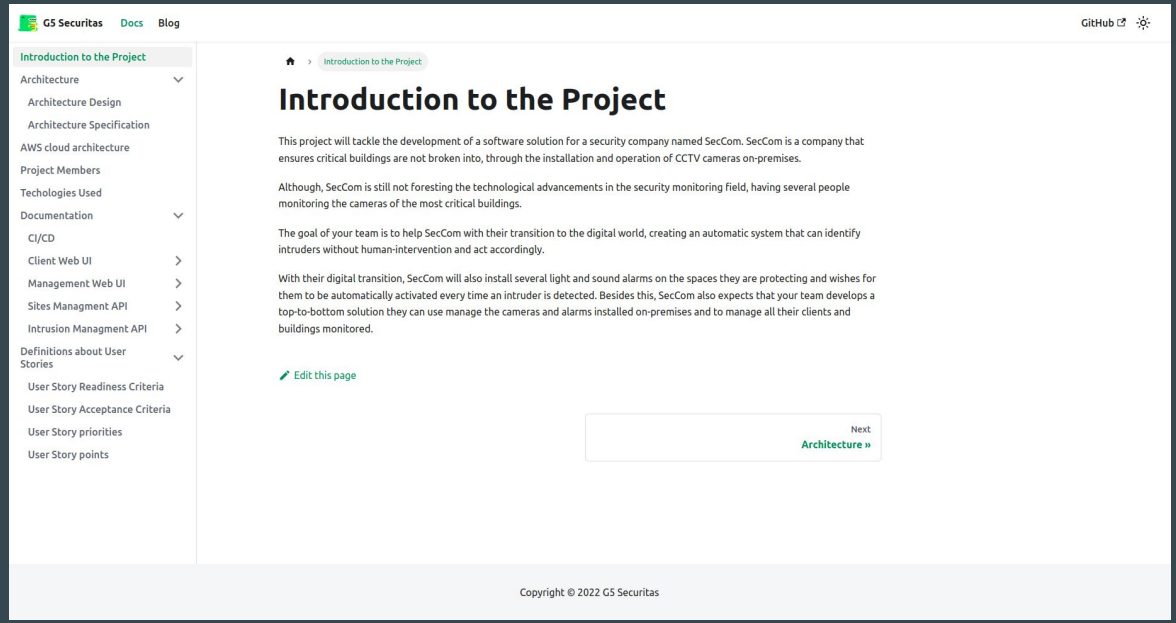
Agile Principles + SCRUM Practices

https://g5-securitas.atlassian.net/jira/software/projects/GS/boards/1



Documentation

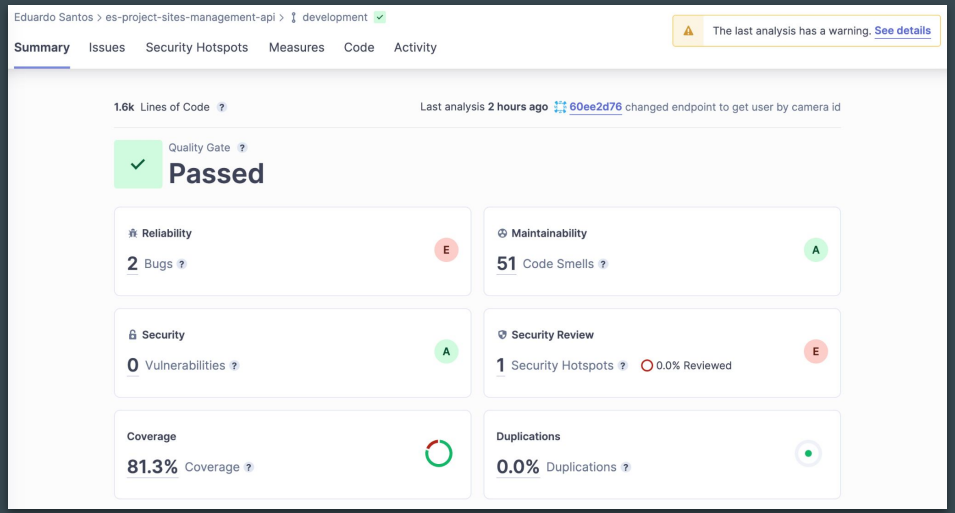
https://eduardosantoshf.github.io/es-project/



CI/CD - Testing

```yaml
name: Sites Management API Coverage and Testing

on:
  push:
    paths:
      - '.github/workflows/sites-management-api-coverage.yaml'
      - sitesManagementAPI/**
      - '!**.md'
      - '!**.pdf'
      - '!**.docx'
      - '!**.gitignore'
      - '!**.txt'
  pull_request:
    paths:
      - sitesManagementAPI/**
      - '!**.md'
      - '!**.pdf'
      - '!**.docx'
      - '!**.gitignore'
      - '!**.txt'
```

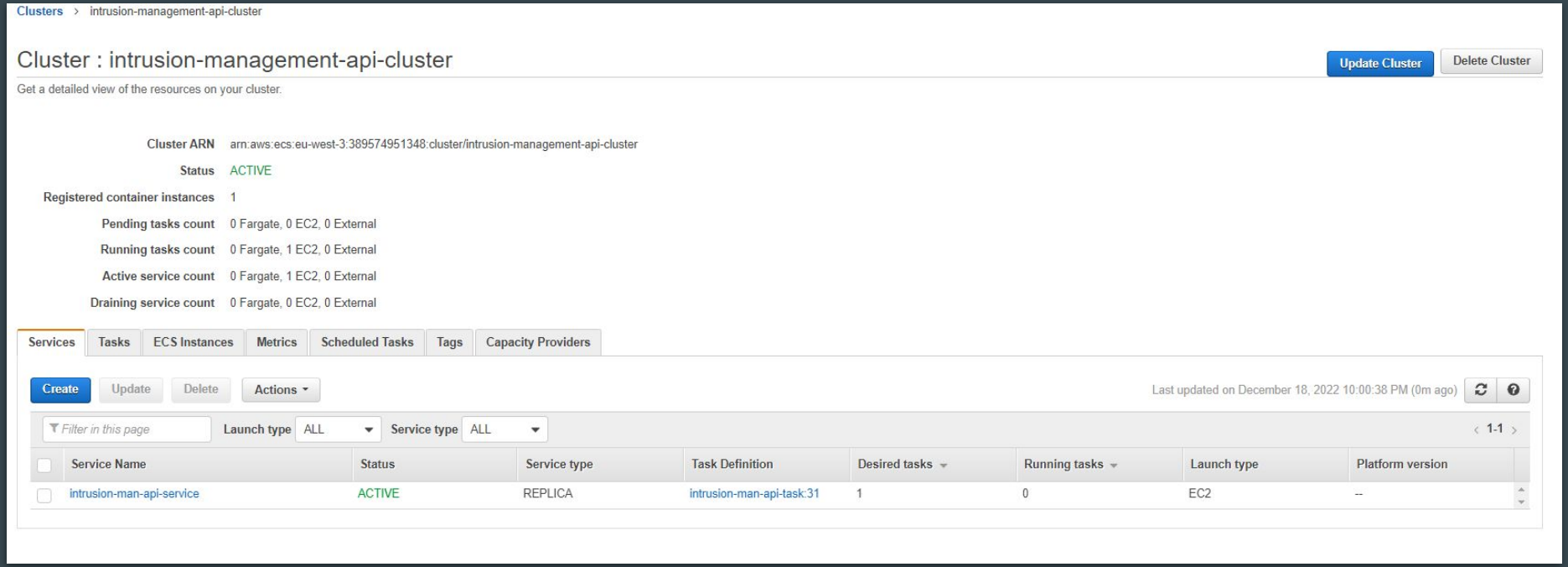



```yaml
steps:
  - name: Pulling git repo
    uses: actions/checkout@v3
    with:
      fetch-depth: 0

  - name: Install tox and any other packages
    run: |
      pip install tox
  - name: Run tox
    run: tox -e py

  - name: SonarCloud Scan
    uses: SonarSource/sonarcloud-github-action@master
    env:
      GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}  # Needed to get PR information, if any
      SONAR_TOKEN: ${{ secrets.SONAR_TOKEN_SITES_MANAGEMENT_API }}
    with:
      projectBaseDir: sitesManagementAPI/
```

CI/CD - Elastic Container Service



CI/CD - Deployment

```yaml
name: Deploy HDM to AWS

on:
  workflow_dispatch:
    inputs:
      logLevel:
        description: 'Log level'
        required: true
        default: 'warning'
        type: choice
        options:
        - info
        - warning
        - debug

  pull_request:
    branches: [ main ]

    paths:
      - HumanDetection/**
      - .github/human-detection-deployment.yaml
      - '!**.md'
      - '!**.pdf'
      - '!**.docx'
      - '!**.gitignore'
      - '!**.txt'

  push:
    branches: [ main ]

    paths:
      - HumanDetection/**
      - .github/human-detection-deployment.yaml
      - '!**.md'
      - '!**.pdf'
      - '!**.docx'
      - '!**.gitignore'
      - '!**.txt'

env:
  ECR_REGISTRY_ALIAS: p3i3q7f7
  ECR_REPOSITORY: human-detection-repository        # set this to your Amazon ECR repository name
  ECS_SERVICE: human-detection-service              # set this to your Amazon ECS service name
  ECS_CLUSTER: human-detection-cluster              # set this to your Amazon ECS cluster name
                                                    # set this to the path to your Amazon ECS task definition
  ECS_TASK_DEFINITION: .aws/human-detection-task-definition.json
                                                    # file, e.g. .aws/task-definition.json
  HDM_CONTAINER_NAME: human-detection-container     # set this to the name of the API container in the
                                                    # containerDefinitions section of your task definition
  RABBIT_MQ_EXCHANGE_NAME: human-detection-exchange
  RABBIT_MQ_QUEUE_NAME: human-detection-queue
```

```yaml
jobs:
  deploy:
    name: Deploy
    runs-on: ubuntu-latest

    steps:
      - name: Checkout
        uses: actions/checkout@v3

      - name: Configure AWS credentials (us-east-1)
        uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
          aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
          aws-region: us-east-1

      - name: Login to Amazon ECR
        id: login-ecr-public
        uses: aws-actions/amazon-ecr-login@v1
        with:
          registry-type: public

      - name: Build, tag, and push docker hdm image to Amazon ECR Public
        id: build-hdm-image
        env:
          ECR_REGISTRY: ${{ steps.login-ecr-public.outputs.registry }}
          IMAGE_TAG: ${{ github.sha }}
          RABBIT_MQ_URL: ${{ secrets.RABBIT_MQ_URL }}
          RABBIT_MQ_USERNAME: ${{ secrets.RABBIT_MQ_USERNAME }}
          RABBIT_MQ_PASSWORD: ${{ secrets.RABBIT_MQ_PASSWORD }}
          REDIS_URL: ${{ secrets.REDIS_URL }}

        run: |
          cd HumanDetection/human-detection-module
          docker build -t $ECR_REGISTRY/$ECR_REGISTRY_ALIAS/$ECR_REPOSITORY:$IMAGE_TAG .
          docker push $ECR_REGISTRY/$ECR_REGISTRY_ALIAS/$ECR_REPOSITORY:$IMAGE_TAG
          echo "image=$ECR_REGISTRY/$ECR_REGISTRY_ALIAS/$ECR_REPOSITORY:$IMAGE_TAG" >> $GITHUB_OUTPUT
      - name: Configure AWS credentials (change to eu-west-3 region)
        uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
          aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
          aws-region: eu-west-3

      - name: Update HDM container in ECS task definition
        id: build-task-def
        uses: aws-actions/amazon-ecs-render-task-definition@v1
        with:
          task-definition: ${{ env.ECS_TASK_DEFINITION }}
          container-name: ${{ env.HDM_CONTAINER_NAME }}
          image: ${{ steps.build-hdm-image.outputs.image }}
          environment-variables: |
            RABBIT_MQ_URL=${{ secrets.RABBIT_MQ_URL }}
            RABBIT_MQ_USERNAME=${{ secrets.RABBIT_MQ_USERNAME }}
            RABBIT_MQ_PASSWORD=${{ secrets.RABBIT_MQ_PASSWORD }}
            RABBIT_MQ_EXCHANGE_NAME=${{ env.RABBIT_MQ_EXCHANGE_NAME }}
            RABBIT_MQ_QUEUE_NAME=${{ env.RABBIT_MQ_QUEUE_NAME }}
            REDIS_URL=${{ secrets.REDIS_URL }}
            INTRUSION_MANAGEMENT_API_URL=${{ env.INTRUSION_MANAGEMENT_API_URL }}
      - name: Deploy Amazon ECS task definition
        uses: aws-actions/amazon-ecs-deploy-task-definition@v1.4.10
        with:
          task-definition: ${{ steps.build-task-def.outputs.task-definition }}
          service: ${{ env.ECS_SERVICE }}
          cluster: ${{ env.ECS_CLUSTER }}
          wait-for-service-stability: false
```

AWS Architecture

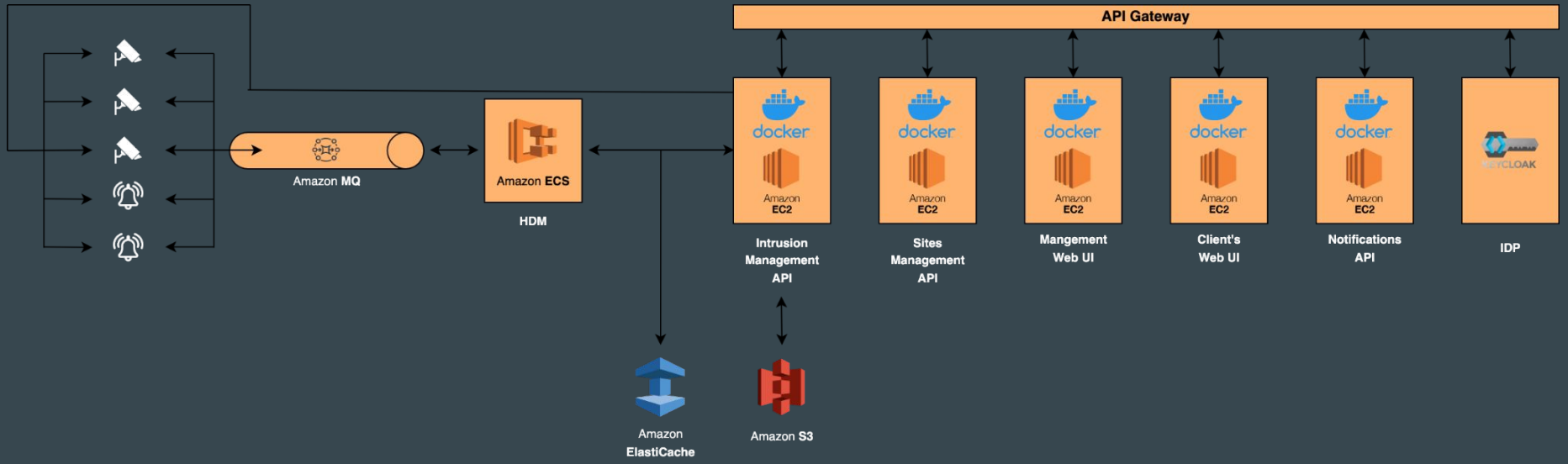
https://eduardosantoshf.github.io/es-project/aws-cloud-architecture/

Security

AA in our system was implemented using an IdP

- Keycloak
- OpenID Connect specification
- Access token

Architecture & Workflow



Solutions' Demo

Client Web UI: http://securitas-lb-1725284772.eu-west-3.elb.amazonaws.com/

Management Web UI: http://ec2-13-37-224-243.eu-west-3.compute.amazonaws.com/admin

Client demo: https://youtu.be/TfNBq4YmH1s

Admin demo: https://youtu.be/hIMHacRem7o

Contributions of Each Team Member

- Diogo Vicente - Sites Management API, IDP, CI/CD
- Eduardo Santos - HDM, CI/CD, Amazon ECS/EC2/MQ
- Eva Bartolomeu - Client UI, Management UI, IDP, Intrusion Management API
- João Ferreira - AWS, CI/CD, Sites Management API
- Pedro Sobral - Cameras, Intrusion Management API, CI, Management UI
- Pedro Bastos - Cameras & HDM, Elastic Cache, AWS Gateway, CI