DevSecOps en aplicaciones Docker y/o Kubernetes

Alumno
Abraham Santana Cebrián

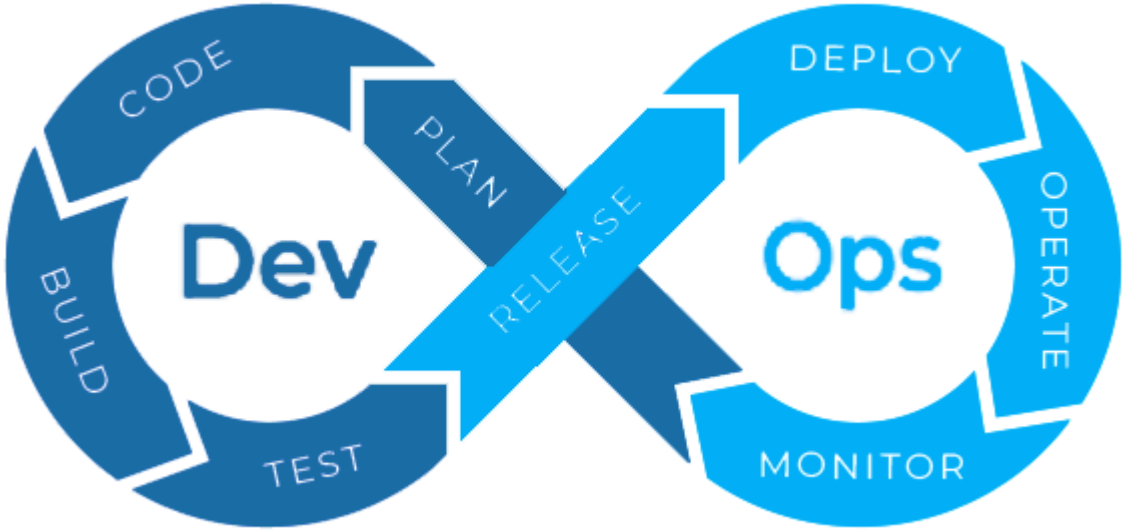
Tutor
Micael Gallego Carrillo

Máster en ciberseguridad y privacidad
Directora
Marta Beltrán Pardo

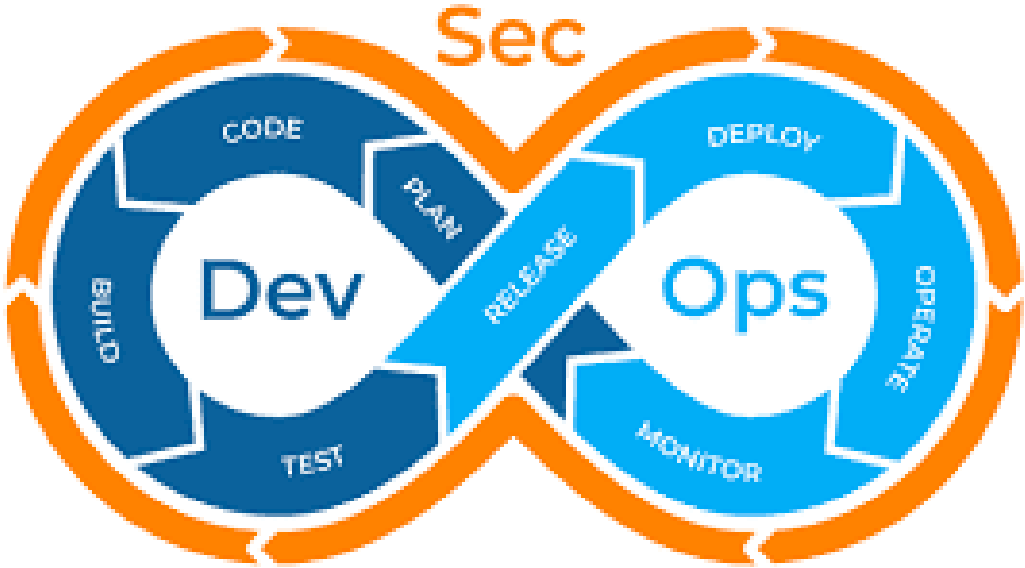
Índice

- Migrar el entorno de Integración Continua / Despliegue Continuo (CI/CD) al proveedor AWS.

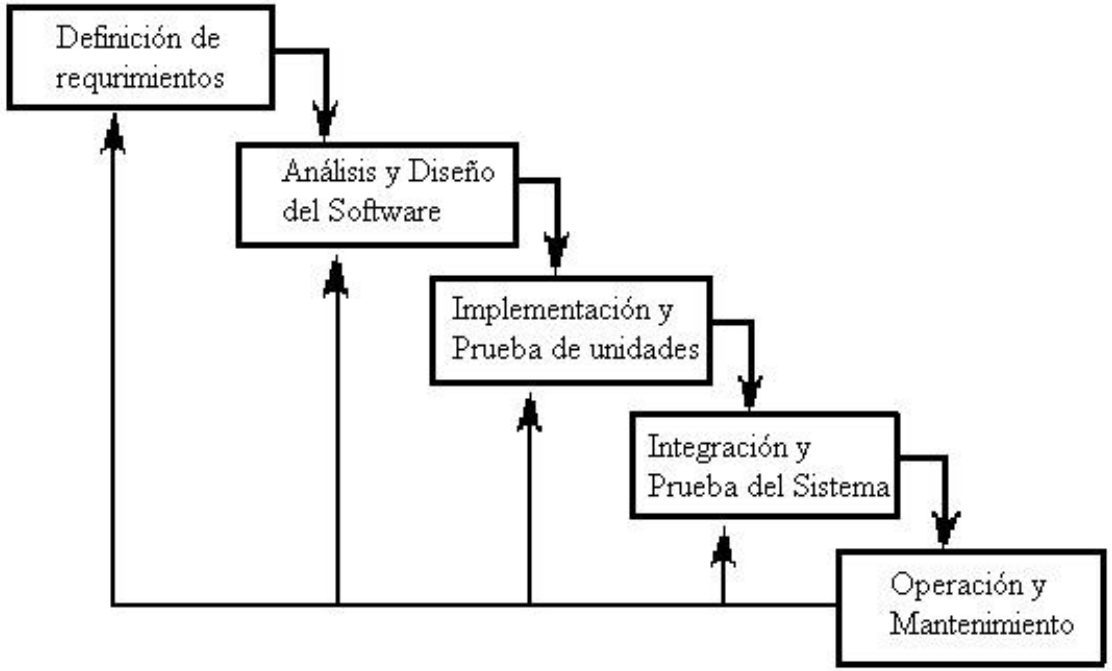
- Desacoplar el entorno CI/CD de una máquina física o virtualizada, ofreciendo las herramientas del entorno, para poder configurar y desplegar en AWS dicho ecosistema.



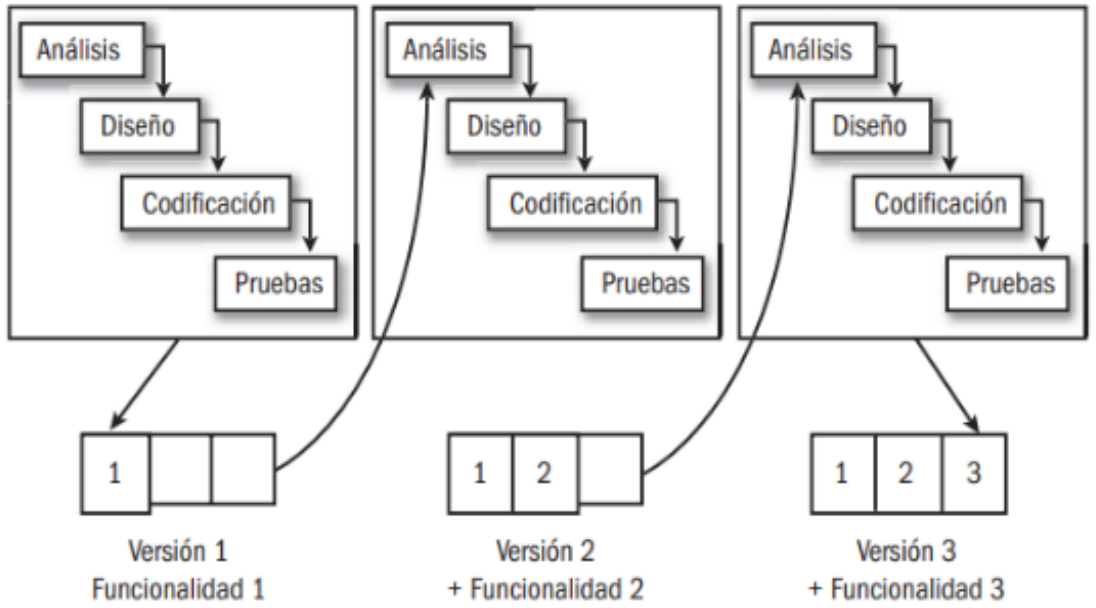
Esquema de funcionamiento de un ciclo DevOps



Esquema de funcionamiento de un ciclo DevSecOps



Modelo cascada realimentado para el ciclo de vida



Modelo iterativo incremental para el ciclo de vida

- **montaje, estudio e implementación de un Pipeline de DevSecOps**, integrando en la implementación todas las herramientas de seguridad posibles, asumiendo que el proyecto de desarrollo software es una aplicación web ya desarrollada , además de posibilitar la implementación de herramientas del tipo análisis estático del código y dependencias), análisis dinámico de código, criterios de análisis de la seguridad, análisis de vulnerabilidades Docker, *pen testing*, etc.

- visión global de las herramientas, de las tecnologías utilizadas, y del sistema de integración de los elementos

- infraestructura configurada a modo de tutorial, explicando cómo se monta un entorno de Continuous Integration / Continuous Deployment (CI/CD), y la configuración de las herramientas utilizadas en este proyecto

```
    - mynet
  volumes:
    - jenkins_home:/var/jenkins_home
    - /var/run/docker.sock:/var/run/docker.sock
    - /usr/local/bin/docker:/usr/local/bin/docker
sonarqube:
  image: sonarqube:lts
  ports:
    - 9000:9000
  networks:
    - mynet
postgres:
  image: postgres:9.6
  restart: unless-stopped
  volumes:
    - ./docker-compose-data/postgres-data/:/var/lib/postgresql/data:rw
  environment:
    - POSTGRES_PASSWORD=ChangeMe
    - POSTGRES_USER=clair
    - POSTGRES_DB=clair
  networks:
    - mynet
clair:
  image: quay.io/coreos/clair:v2.0.6
  command: -config=/config/config.yaml
  #command: -log-level=debug
  ports:
    - 6060:6060
    - 6061:6061
  depends_on:
    - postgres
  volumes:
    - ./docker-compose-data/clair-config/:/config/:ro
    - ./docker-compose-data/clair-tmp/:/tmp/:rw
  user: root
  networks:
```

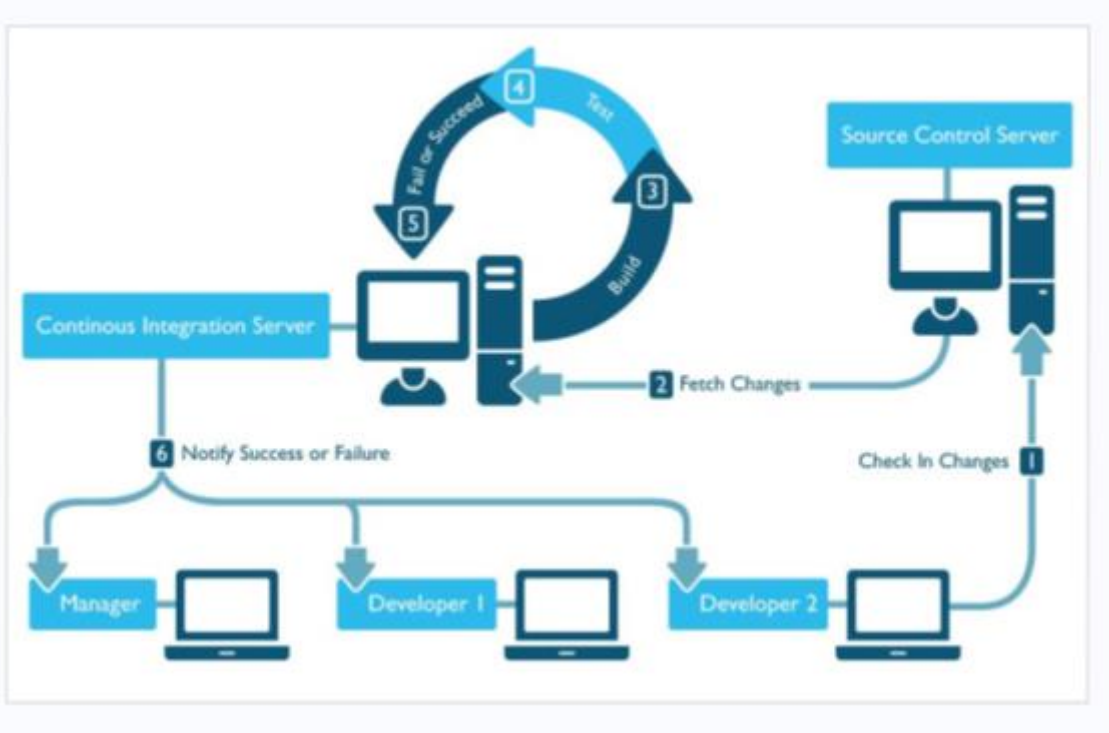
Animación con el script del Pipeline y YAMLD Docker-Compose

Jenkins

docker

sonarqube

DEPENDENCY-CHECK



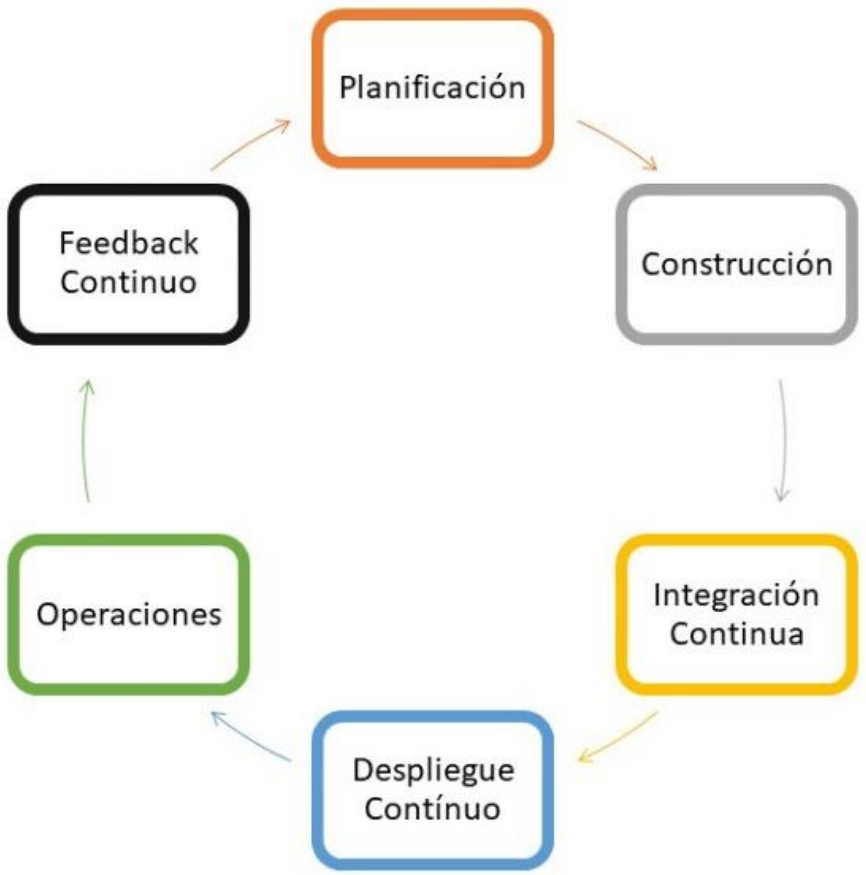
Esquema conceptual de Integración Continua y Despliegue Continuo

clair

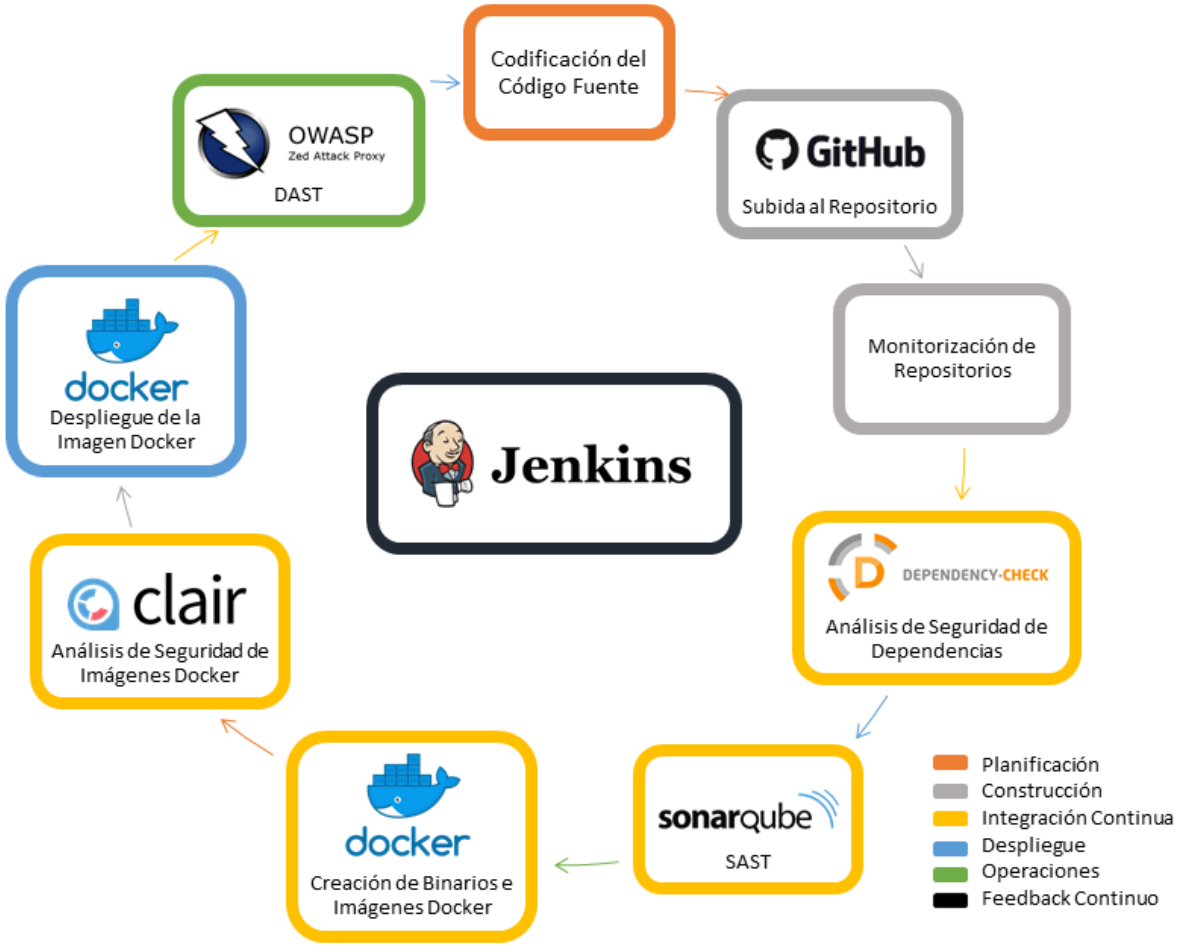
WEBGOAT

OWASP
Zed Attack Proxy

kubernetes

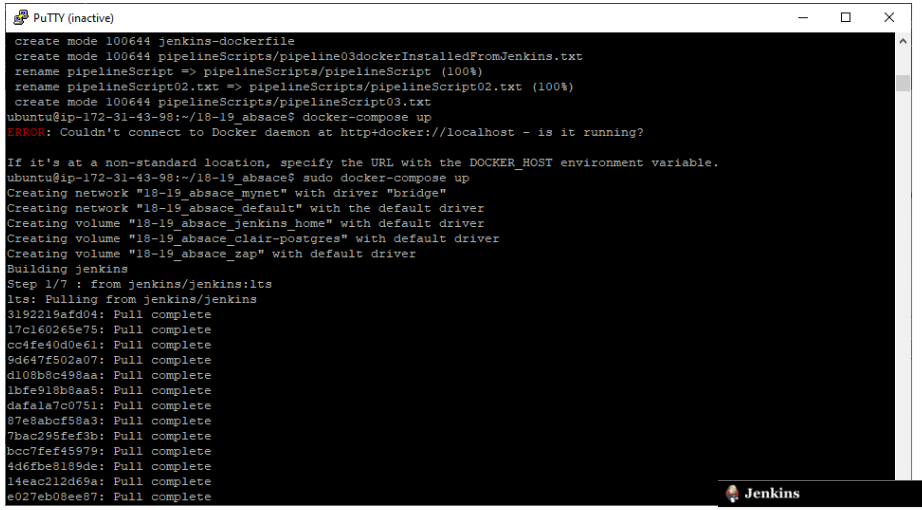


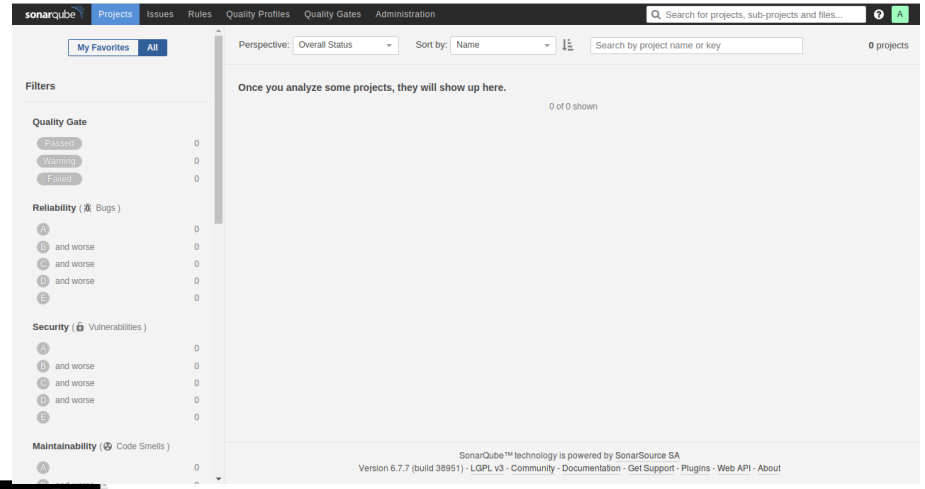
Etapas generales en DevOps

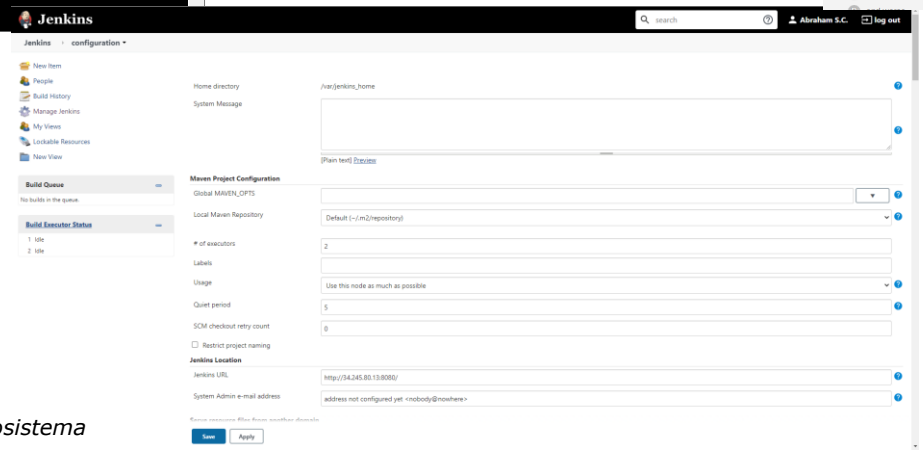


Etapas DevSecOps en este Trabajo

- Docker-Compose: orquestación del entorno DevSecOps
- Jenkins: configuración y puesta a punto
- SonarQube: configuración y puesta a punto

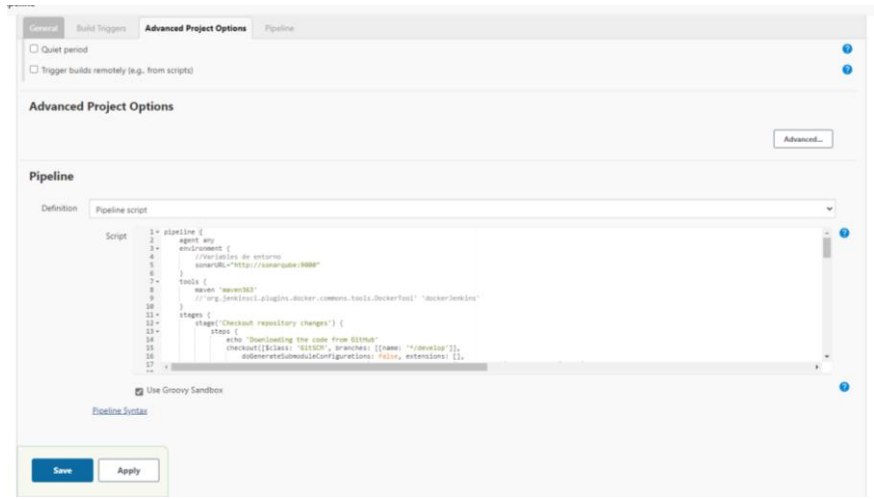




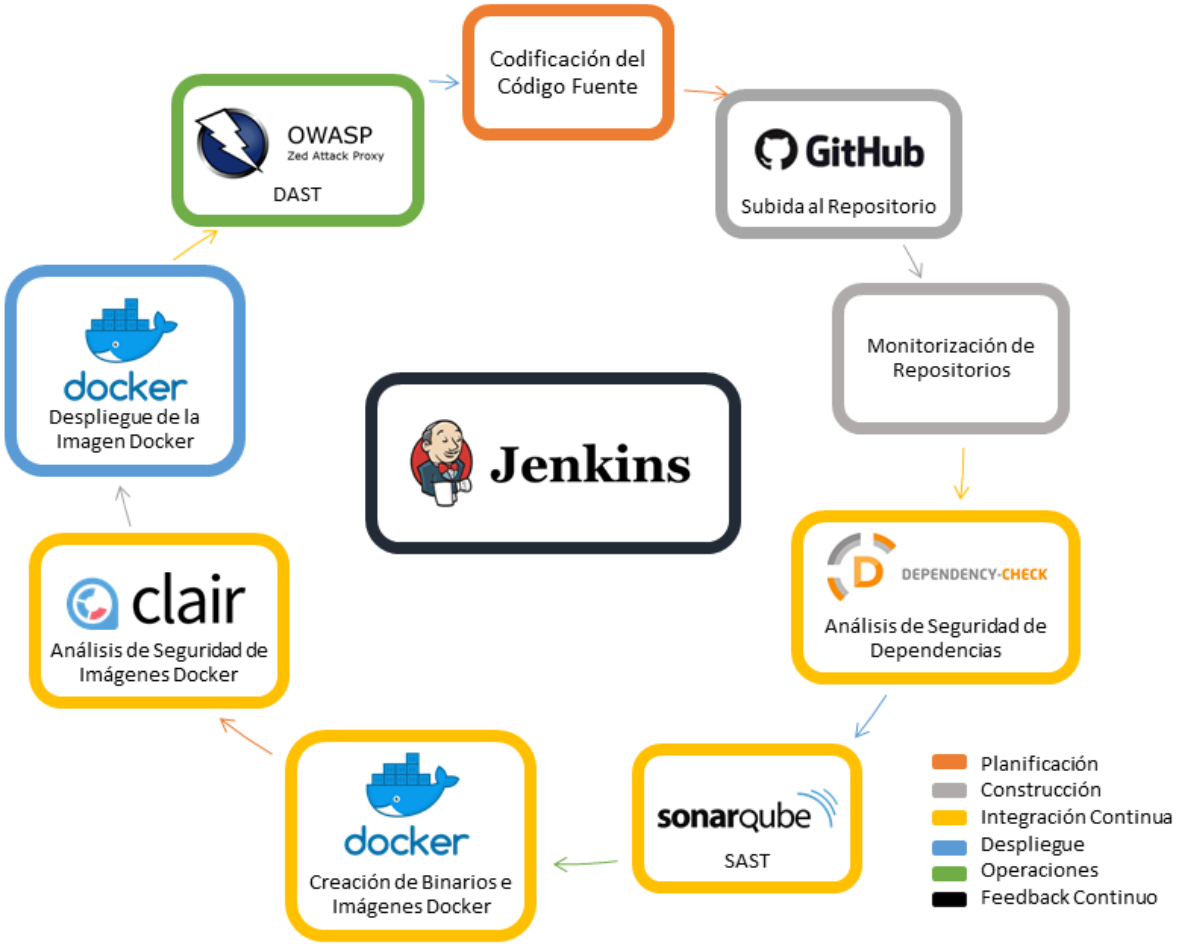


Imágenes de los resultados obtenidos en la implementación del ecosistema basado desplegado en Docker

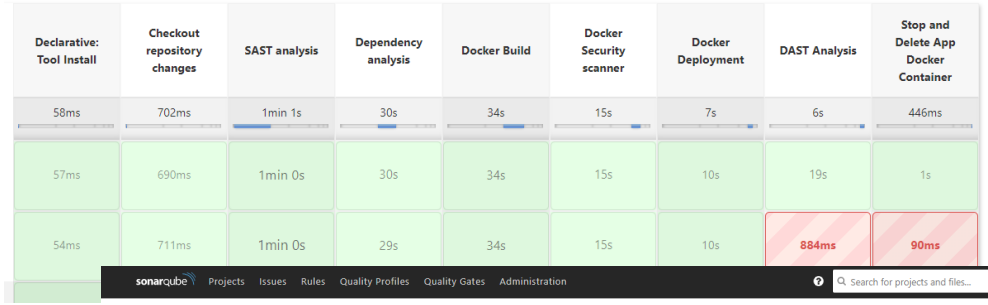
• Configuración del Pipeline

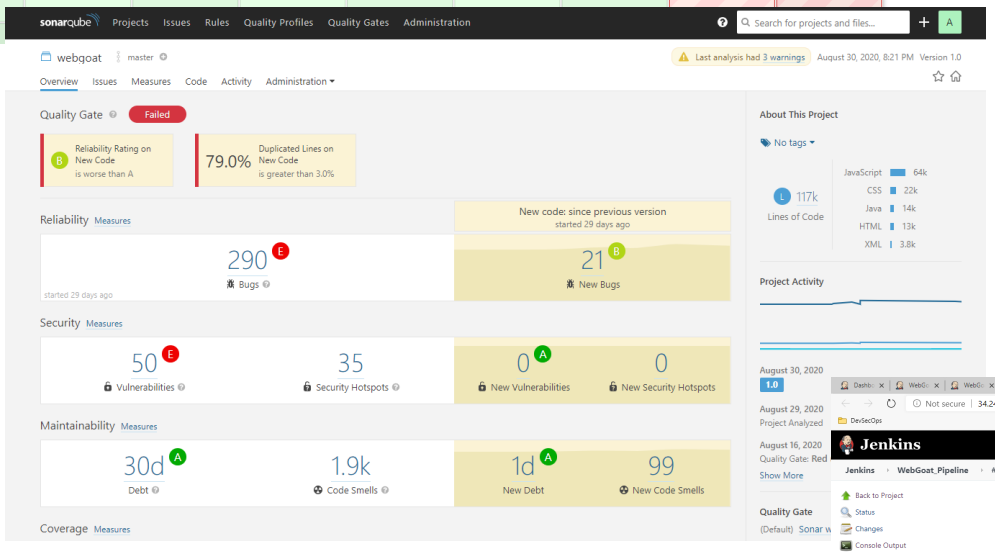


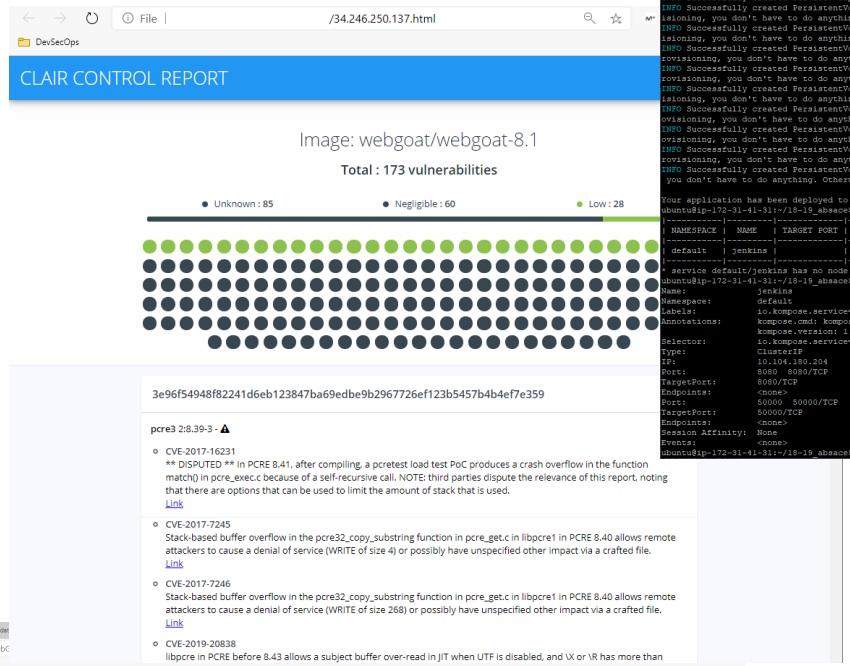
• Etapas del Pipeline

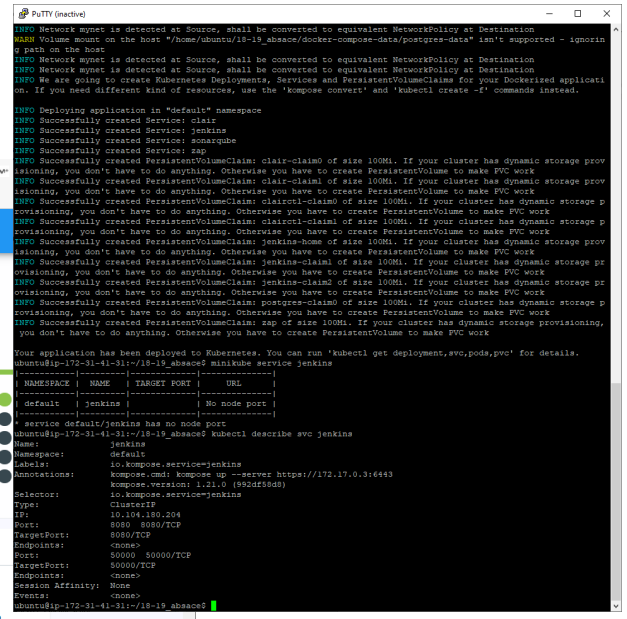


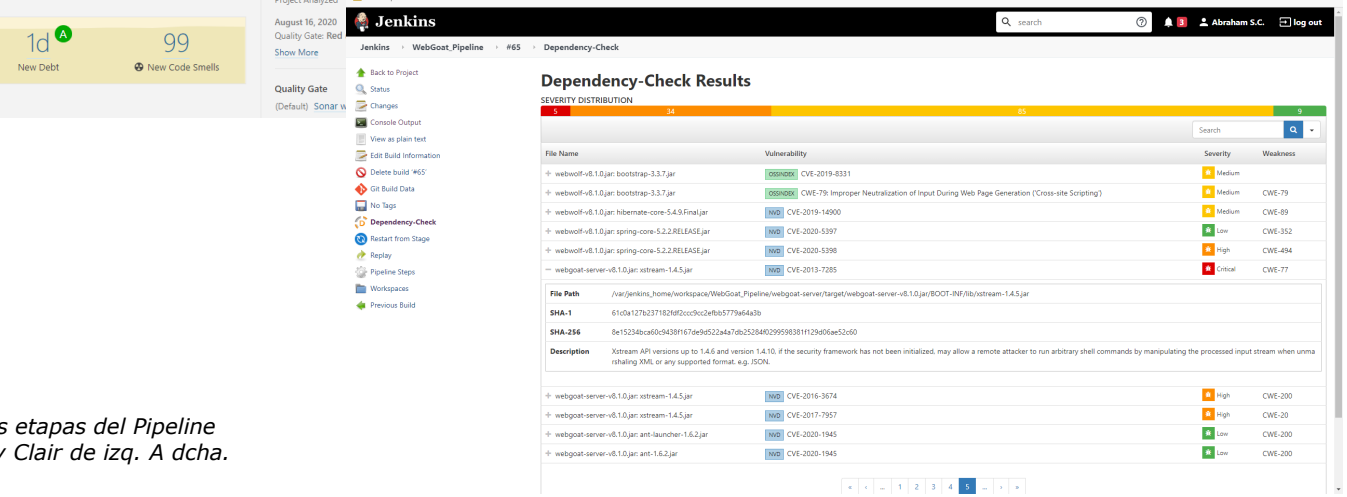
Imágenes de los resultados obtenidos durante la codificación de las pruebas.







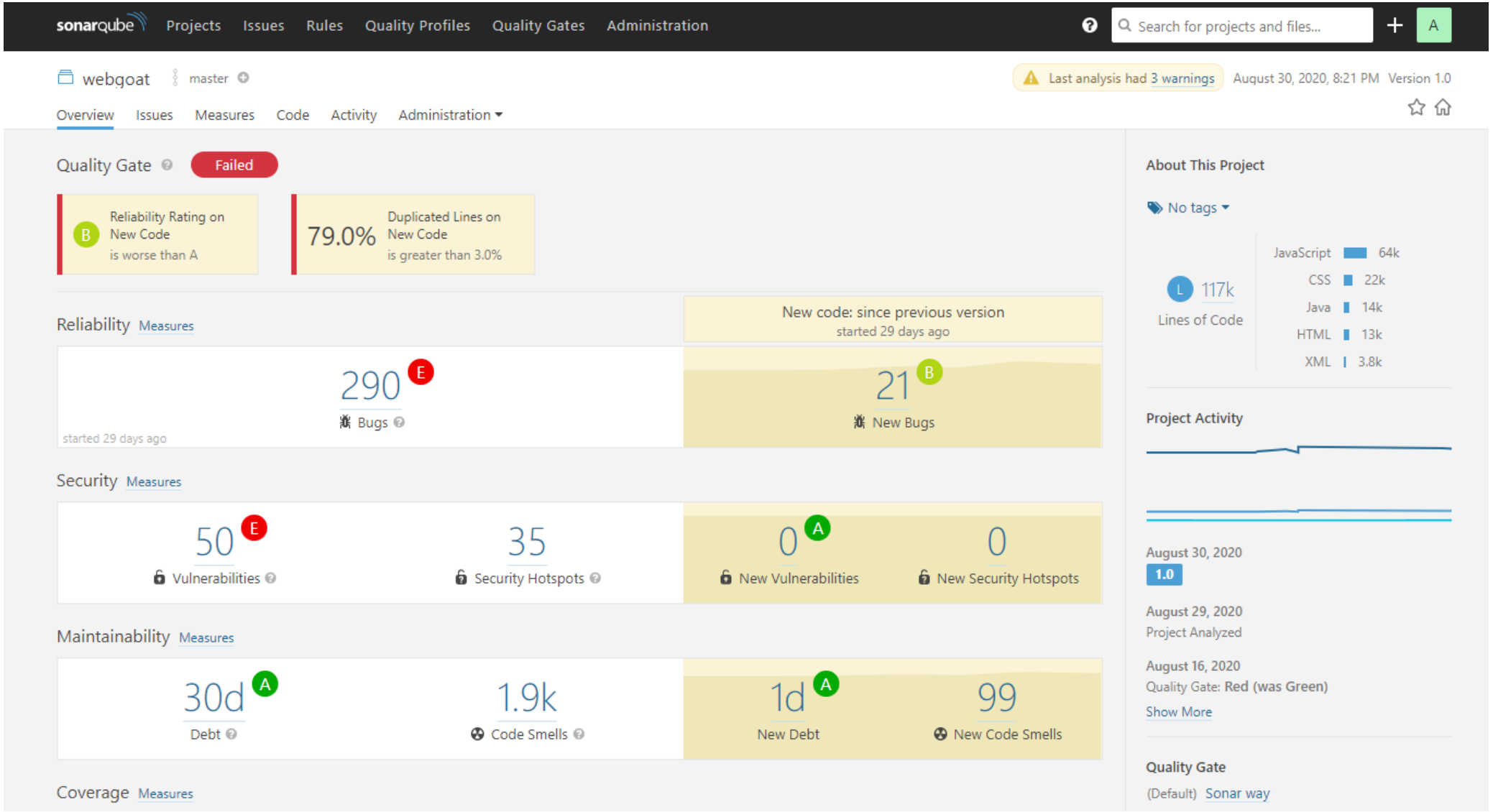




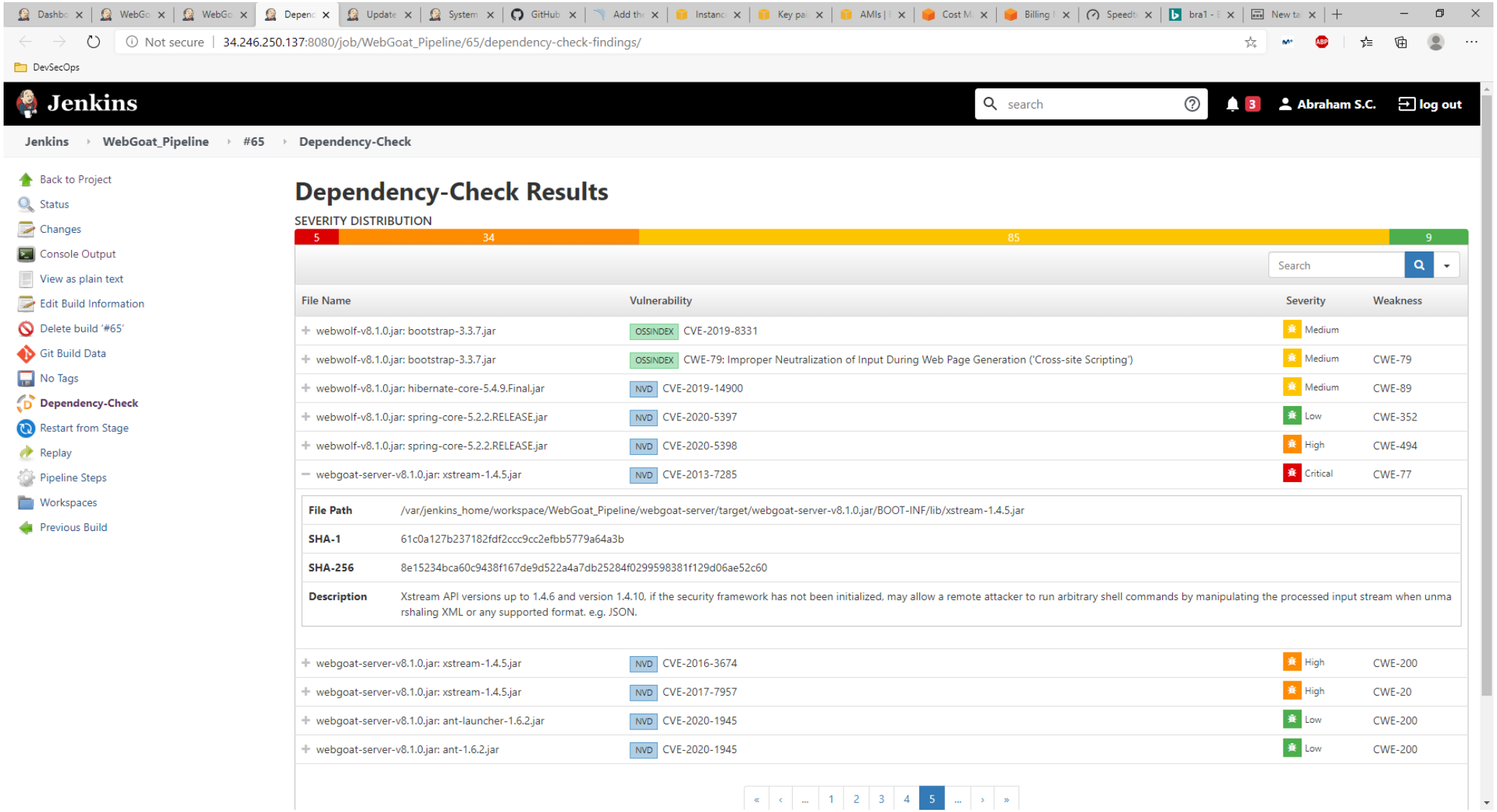
Imágenes de los resultados obtenidos en las diferentes etapas del Pipeline DevSecOps: Jenkins, Sonarqube, Dependency-Check y Clair de izq. A dcha.

Declarative: Tool Install	Checkout repository changes	SAST analysis	Dependency analysis	Docker Build	Docker Security scanner	Docker Deployment	DAST Analysis	Stop and Delete App Docker Container
58ms	702ms	1min 1s	30s	34s	15s	7s	6s	446ms
57ms	690ms	1min 0s	30s	34s	15s	10s	19s	1s
54ms	711ms	1min 0s	29s	34s	15s	10s	884ms failed	90ms failed

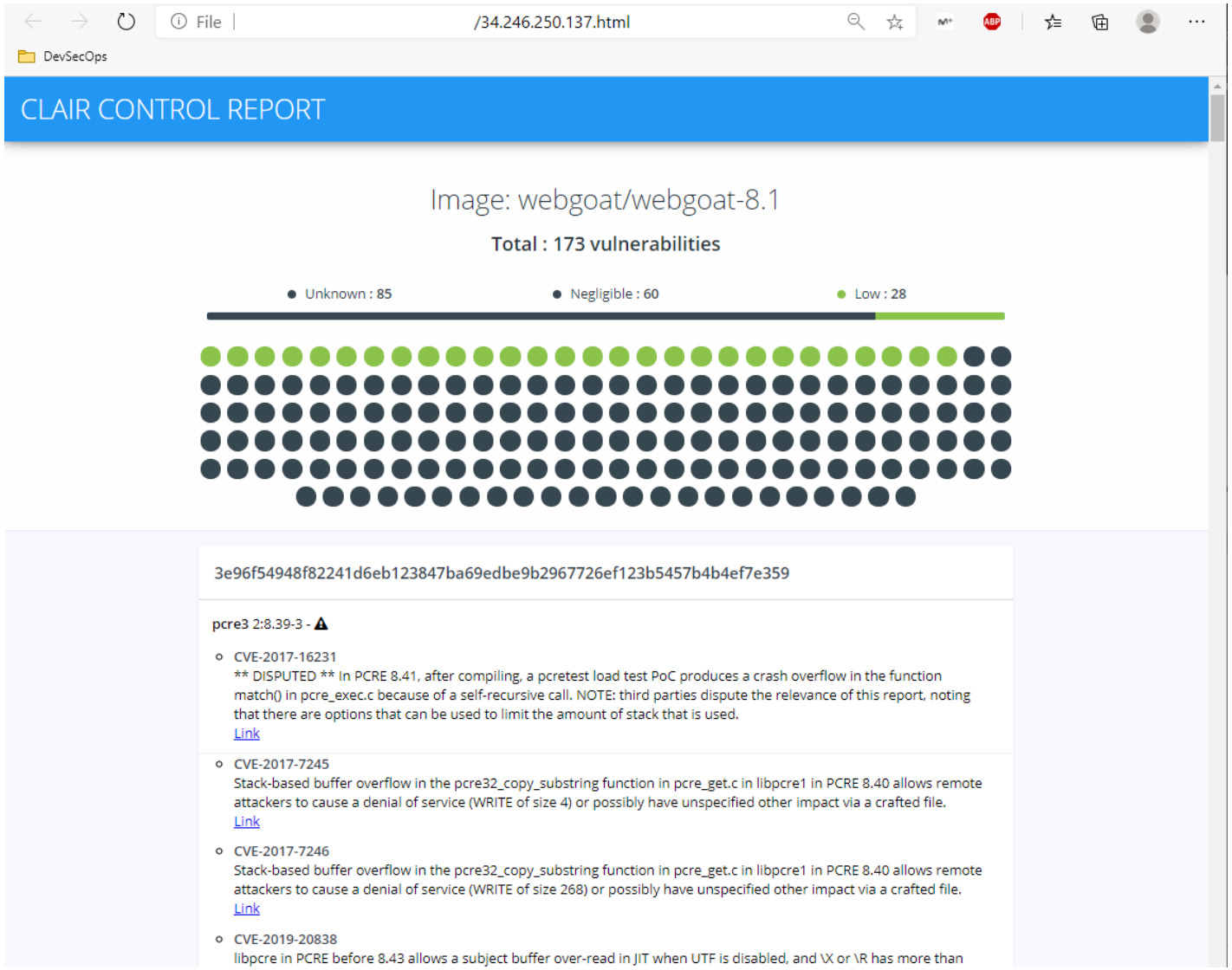
Imágenes de los resultados obtenidos en las diferentes etapas del Pipeline DevSecOps: Jenkins, Sonarqube, Dependency-Check y Clair de izq. A dcha.



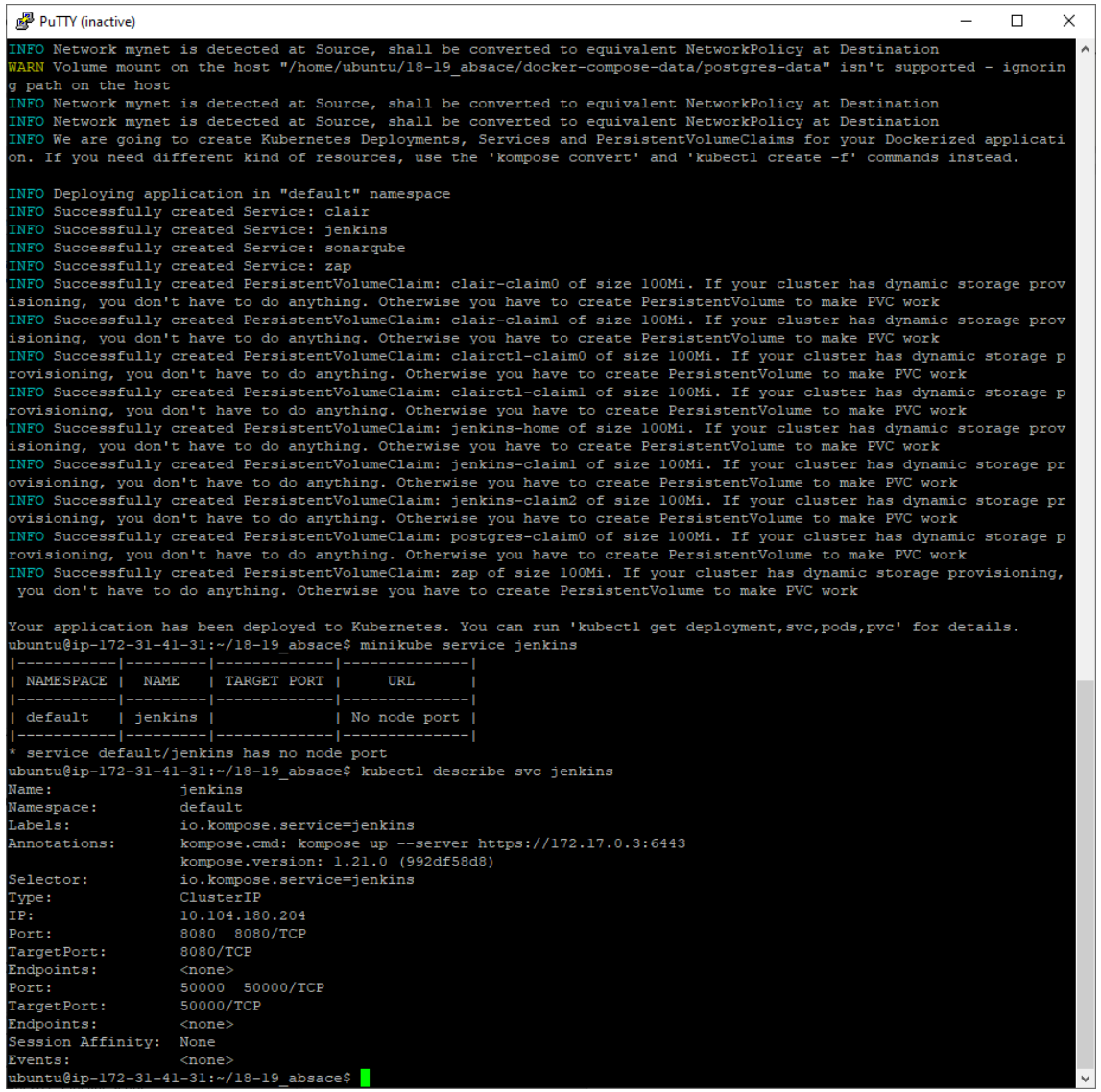
Imágenes de los resultados obtenidos en las diferentes etapas del Pipeline DevSecOps: Jenkins, Sonarqube, Dependency-Check y Clair de izq. A dcha.



Imágenes de los resultados obtenidos en las diferentes etapas del Pipeline DevSecOps: Jenkins, Sonarqube, Dependency-Check y Clair de izq. A dcha.



Imágenes de los resultados obtenidos en las diferentes etapas del Pipeline DevSecOps: Jenkins, Sonarqube, Dependency-Check y Clair de izq. A dcha.



Imágenes de los resultados obtenidos en las diferentes etapas del Pipeline DevSecOps: Jenkins, Sonarqube, Dependency-Check y Clair de izq. A dcha.

- Implementación de la herramienta Docker dentro de la imagen y contenedor Jenkins utilizado en este Trabajo

- Error de codificación del fichero Dockerfile para la empaquetación de la app WebGoat en una imagen Docker

- Fichero de configuración en Clair e integración de Clair y Clairctl en el ecosistema DevSecOps dockerizado

- Incompatibilidades en la migración del ecosistema DevSecOps hacia Kubernetes utilizando la herramienta Kompose

- implementar un sistema CI/CD teniendo en cuenta los requisitos y las políticas de seguridad, así como la implementación de seguridad SAST, DAST, análisis de dependencias, análisis de imágenes Docker, nos proporcionan un nivel de información muy elevado sobre la calidad del producto software que se está desplegando, así como los riesgos de seguridad que tiene su entrega al mercado

- la capacidad de recursos del Host donde se despliegue este sistema es limitante

- el desarrollo de un proyecto software con un Pipeline DevSecOps será mucho más sencilla que con un simple DevOps si se logra integrar herramientas en flujo de Pipeline de Jenkins

- la importancia de añadir la seguridad en todas las etapas, resaltando la idea de que seguridad y desarrollo son incompatibles

```
    - mynet
  volumes:
    - jenkins_home:/var/jenkins_home
    - /var/run/docker.sock:/var/run/docker.sock
    - /usr/local/bin/docker:/usr/local/bin/docker
sonarqube:
  image: sonarqube:lts
  ports:
    - 9000:9000
  networks:
    - mynet
postgres:
  image: postgres:9.6
  restart: unless-stopped
  volumes:
    - ./docker-compose-data/postgres-data/:/var/lib/postgresql/data:rw
  environment:
    - POSTGRES_PASSWORD=ChangeMe
    - POSTGRES_USER=clair
    - POSTGRES_DB=clair
  networks:
    - mynet
clair:
  image: quay.io/coreos/clair:v2.0.6
  command: -config=/config/config.yaml
  #command: -log-level=debug
  ports:
    - 6060:6060
    - 6061:6061
  depends_on:
    - postgres
  volumes:
    - ./docker-compose-data/clair-config/:/config/:ro
    - ./docker-compose-data/clair-tmp/:/tmp/:rw
  user: root
  networks:
```

- Realizar un estudio y la integración adecuada de la herramienta Notary, la cual se utiliza para analizar y asegurar que las imágenes Docker que se están utilizando han sido firmadas, comprobando así su integridad, confidencialidad, y disponibilidad.

notary

- Realizar un estudio e implementación más completo del realizado en este trabajo para el análisis DAST utilizando la herramienta OWASP ZAP, ya que no se han conseguido resultados lo suficientemente claros en este Trabajo.

OWASP
Zed Attack Proxy

- Realizar un estudio y mejora continua del script del Pipeline, ya siempre es posible encontrar mejoras en el rendimiento y funcionalidad de este, siguiendo la metodología DevSecOps.

Jenkins

- Realizar un estudio necesario para la migración y adaptación de todo el ecosistema DevSecOps creado en este Trabajo a un entorno Kubernetes, analizando la viabilidad de hacer uso de la herramienta Kompose u otras herramientas.

kubernetes

DevSecOps en aplicaciones Docker y/o Kubernetes

Demo del ecosistema y ejecución del Pipeline DevSecOps

https://youtu.be/HkFP6tAobp0

¿Preguntas?

a.santanac.2018@alumnos.urjc.es

Alumno
Abraham Santana Cebrián

Tutor
Micael Gallego Carrillo

Máster en ciberseguridad y privacidad
Directora
Marta Beltrán Pardo

DevSecOps en aplicaciones Docker y/o Kubernetes

Muchas gracias por su atención

Alumno
Abraham Santana Cebrián

Tutor
Micael Gallego Carrillo

Máster en ciberseguridad y privacidad
Directora
Marta Beltrán Pardo