

¿Qué es y para que se usa la integración continua?

Es una práctica de ingeniería de software que consiste en hacer integraciones automáticas de un proyecto lo más a menudo posible para así poder detectar fallos cuanto antes.

Entendemos por integración la compilación y ejecución de pruebas de todo un proyecto.

Gitea + Drone + SonarQube

Se utiliza el software libre de Gitea para almacenar los *repositorios de código* y que conectaremos a Drone para realizar la IC

Drone permite ejecutar las *pipelines* que tiene configuradas el repositorio, cuenta con mucha variedad de plugins dando posibilidad de despliegues o comunicaciones

SonarQube provee de métricas y analiza la calidad y seguridad del código

Con este stack podremos montar un ciclo de IC que cubre las necesidades básicas, también destaca los pocos recursos que necesita esta configuración pudiendo desplegarse en máquinas pequeñas como una Raspberry

Docker

Usamos Docker para desplegar las aplicaciones en distintos contenedores

```
version: '2'

services:
  gitea:
    image: gitea/gitea
    restart: always
    container_name: gitea
    hostname: gitea
    volumes:
      - ./data:/data
    expose:
      - 3000
    depends_on:
      - db
    networks:
      mynet:

  db:
    image: mariadb:10
    container_name: gitea-db
    hostname: gitea-db
    restart: always
    environment:
      - MYSQL_ROOT_PASSWORD=secretdb
      - MYSQL_DATABASE=gitea
      - MYSQL_USER=gitea
      - MYSQL_PASSWORD=secretpass
    volumes:
      - ./db:/var/lib/mysql
    networks:
      mynet:
```

```
drone:
  image: drone/drone
  restart: always
  container_name: drone
  hostname: drone
  environment:
    - DRONE_GITEA_SERVER=https://gitea.domain.com
    - DRONE_GITEA_CLIENT_ID=secretid
    - DRONE_GITEA_CLIENT_SECRET=secret
    - DRONE_RPC_SECRET=secret-rpc
    - DRONE_SERVER_HOST=drone.domain.com
    - DRONE_SERVER_PROTO=https
  volumes:
    - ./drone:/data
  expose:
    - 80
  networks:
    mynet:

runner:
  image: drone/drone-runner-docker
  restart: always
  container_name: runner
  hostname: runner
  environment:
    - DRONE_RPC_PROTO=http
    - DRONE_RPC_HOST=drone
    - DRONE_RPC_SECRET=secret-rpc
    - DRONE_RUNNER_CAPACITY=2
    - DRONE_RUNNER_NAME=runner
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
```

```
  expose:
    - 3000
  networks:
    mynet:

sonarqube:
  image: sonarqube
  container_name: sonarqube
  hostname: sonarqube
  expose:
    - 9000
  volumes:
    - ./sonar/conf:/opt/sonarqube/conf
    - ./sonar/data:/opt/sonarqube/data
    - ./sonar/logs:/opt/sonarqube/logs
    - ./sonar/extensions:/opt/sonarqube/extensions
  networks:
    mynet:

networks:
  mynet:
```

Solo hay que crear la app en Gitea para conectarla a Drone y generar las claves de la configuración del *runner* de Drone. También habría que generar la token del plugin de sonar para cada proyecto de Drone que configuremos

Con esta receta de *docker-compose* se puede poner un *proxy* que ataque los distintos contenedores bajo el mismo dominio.com

Eso es todo, nos vemos en la próxima