

# Herramientas de automatización



Facultad de Ingeniería  
Escuela de Ingeniería de Sistemas y Computación

# CONCEPTOS

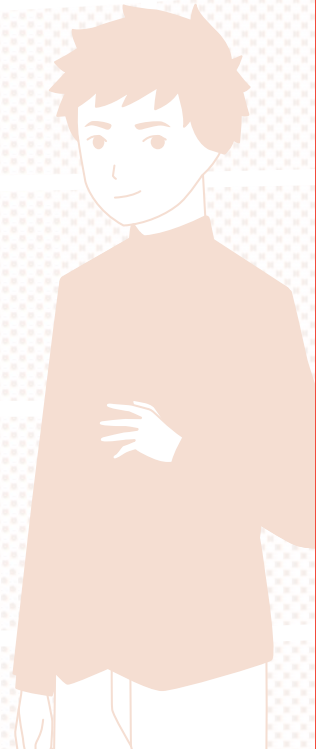
- Jenkins
- Github actions

# AUTOMATIZACIÓN

La automatización es esencial en DevOps porque:

- Acelera la entrega del software.
- Mejora la calidad del software.
- Aumenta la eficiencia y reduce costos.
- Incrementa la colaboración entre los equipos.
- Facilita la escalabilidad del negocio.

En definitiva, la automatización en DevOps ayuda a las empresas a ser más ágiles, eficientes y efectivas en la entrega de software y en la gestión de sus operaciones.



# HERRAMIENTAS



Herramienta	Descripción	Leng programación	Integraciones
Jenkins	Herramienta de integración continua y entrega continua (CI/CD) de código abierto.	JAVA	AWS, Azure, Docker, GitHub, JIRA, entre otros.
Travis CI	Plataforma de integración continua y entrega continua (CI/CD) para proyectos de código abierto y privados.	RUBY, YAML	Github
GitHub Actions	Herramienta de integración continua y entrega continua (CI/CD) de GitHub.	YAML	GitHub
CircleCI	Herramienta de integración continua y entrega continua (CI/CD) integrada en GitLab.	YAML	GitHub, Bitbucket, GitLab, entre otros.



# JENKINS

Jenkins proporciona una gran cantidad de plugins y herramientas para integrarse con diferentes sistemas, lo que hace que sea fácil de configurar y personalizar según las necesidades de cada proyecto y equipo. Además, Jenkins tiene una gran comunidad de usuarios y desarrolladores que contribuyen a su evolución constante y lo hacen una de las herramientas más populares en el mundo del desarrollo de software.






# JENKINS


- Panel de control: proporciona una vista general del estado de los trabajos de construcción y permite la configuración de nuevas tareas.
- Consola de registro: muestra información detallada de los resultados de las tareas de construcción, incluidos los errores y las advertencias.
- Integración de control de versiones: proporciona integración con sistemas de control de versiones como Git, SVN, Mercurial, entre otros.
- Integración de herramientas: proporciona integración con herramientas de construcción como Docker, Ant, Maven, Gradle, entre otros.
- Plugins: Jenkins cuenta con una gran cantidad de plugins disponibles que permiten extender y personalizar su funcionalidad.

# JENKINSFILE

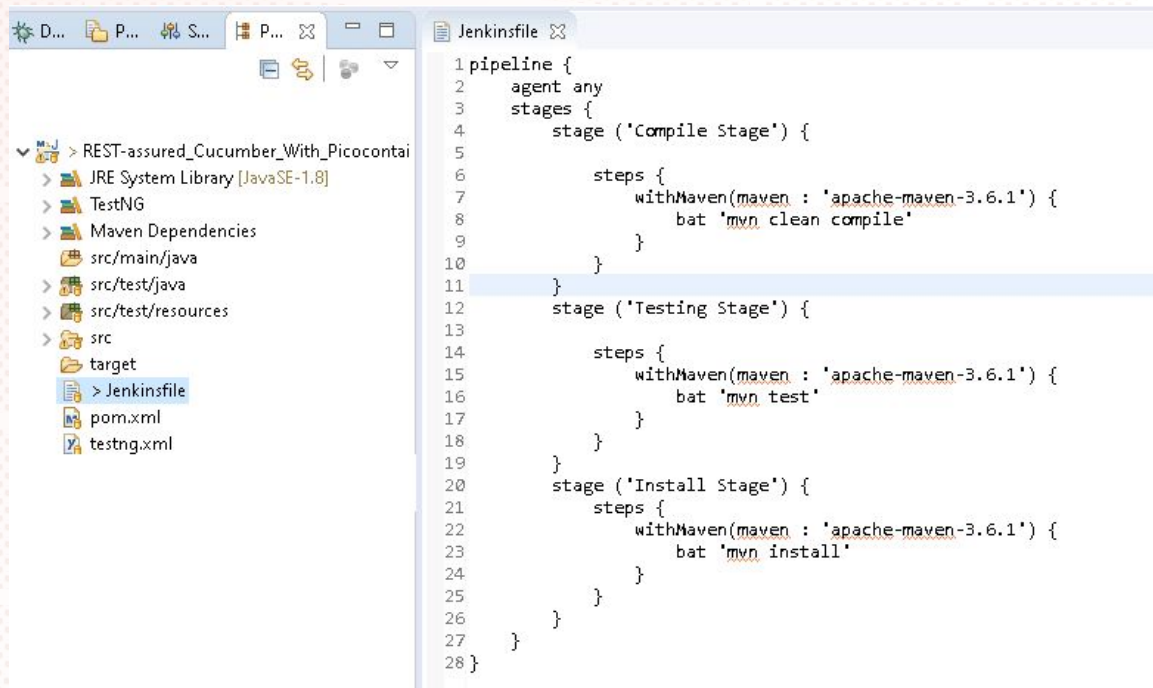


Permite describir de forma programática y versionada el flujo de trabajo completo de construcción, prueba y despliegue de una aplicación.

Al utilizar un Jenkinsfile, se puede definir el proceso de integración de pruebas como código, lo que lo hace más escalable, mantenible y fácilmente compartible entre equipos. También permite automatizar tareas repetitivas y notificar a los miembros del equipo cuando falla alguna etapa del pipeline.



# JENKINSFILE



The screenshot shows an IDE with two panels. The left panel displays a project structure for 'REST-assured\_Cucumber\_With\_Picocontai'. The right panel shows the content of the 'Jenkinsfile'.

**Project Structure (Left Panel):**

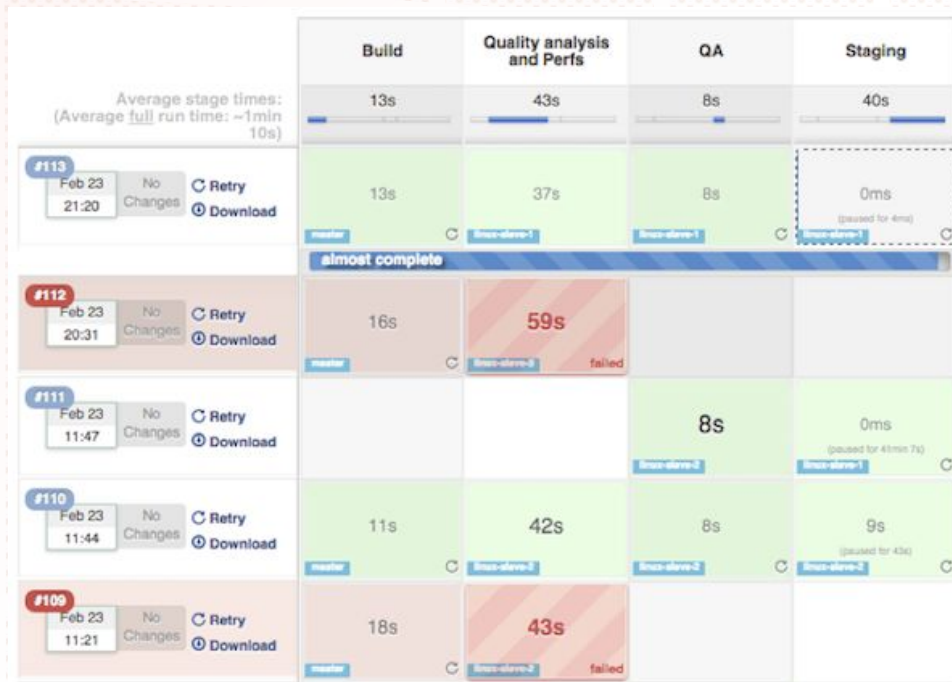
- REST-assured\_Cucumber\_With\_Picocontai
  - JRE System Library [JavaSE-1.8]
  - TestNG
  - Maven Dependencies
  - src/main/java
  - src/test/java
  - src/test/resources
  - src
    - target
    - Jenkinsfile
    - pom.xml
    - testng.xml

**Jenkinsfile Content (Right Panel):**

```
1 pipeline {
2   agent any
3   stages {
4     stage ('Compile Stage') {
5
6       steps {
7         withMaven(maven : 'apache-maven-3.6.1') {
8           bat 'mvn clean compile'
9         }
10      }
11    }
12    stage ('Testing Stage') {
13
14      steps {
15        withMaven(maven : 'apache-maven-3.6.1') {
16          bat 'mvn test'
17        }
18      }
19    }
20    stage ('Install Stage') {
21      steps {
22        withMaven(maven : 'apache-maven-3.6.1') {
23          bat 'mvn install'
24        }
25      }
26    }
27  }
28 }
```



# WORKFLOW



# JENKINS

¿QUÉ ES?



# GITHUB ACTIONS

Plataforma de automatización de flujo de trabajo integrada dentro de la plataforma de alojamiento de código fuente GitHub.

Permite definir y ejecutar flujos de trabajo de forma automatizada, en respuesta a eventos específicos, como la apertura de una pull request, la confirmación de cambios en una rama, o la publicación de una nueva versión. Los flujos de trabajo se definen mediante archivos YAML, que describen las tareas que deben ejecutarse, los entornos en los que deben ejecutarse y las condiciones en las que deben activarse.

GitHub Actions permite integrarse con otros servicios externos y herramientas, como Docker, AWS, Microsoft Azure, Google Cloud, entre otros, lo que permite automatizar procesos de compilación, pruebas, empaquetado, despliegue y notificación.



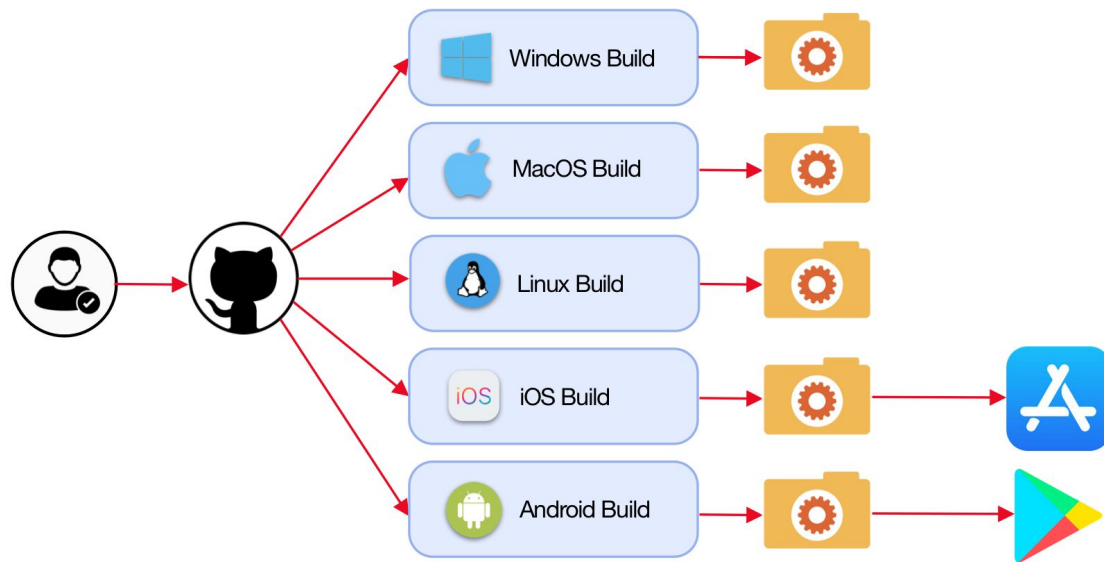
GitHub Actions

# GITHUB ACTIONS CONCEPTOS

- **Acciones:** las acciones son tareas individuales que se pueden combinar para crear flujos de trabajo personalizados.
- **Flujos de trabajo:** los flujos de trabajo son procesos automatizados que se activan en respuesta a eventos específicos, como una confirmación de cambios o una publicación de código fuente.
- **Archivos YAML:** los flujos de trabajo se definen utilizando archivos YAML, que describen las tareas que deben ejecutarse, los entornos en los que deben ejecutarse y las condiciones en las que deben activarse.
- **Ambientes:** los ambientes son entornos de ejecución que pueden configurarse para que las tareas se ejecuten en diferentes plataformas, como sistemas operativos, lenguajes de programación y servicios externos.
- **Secretos:** los secretos son datos confidenciales, como contraseñas o claves de API, que se pueden almacenar de forma segura en el repositorio y utilizar en flujos de trabajo sin exponerlos públicamente.
- **Eventos:** los eventos son acciones específicas que ocurren en el repositorio, como la apertura de una solicitud de extracción o la confirmación de un cambio, que pueden activar flujos de trabajo específicos.

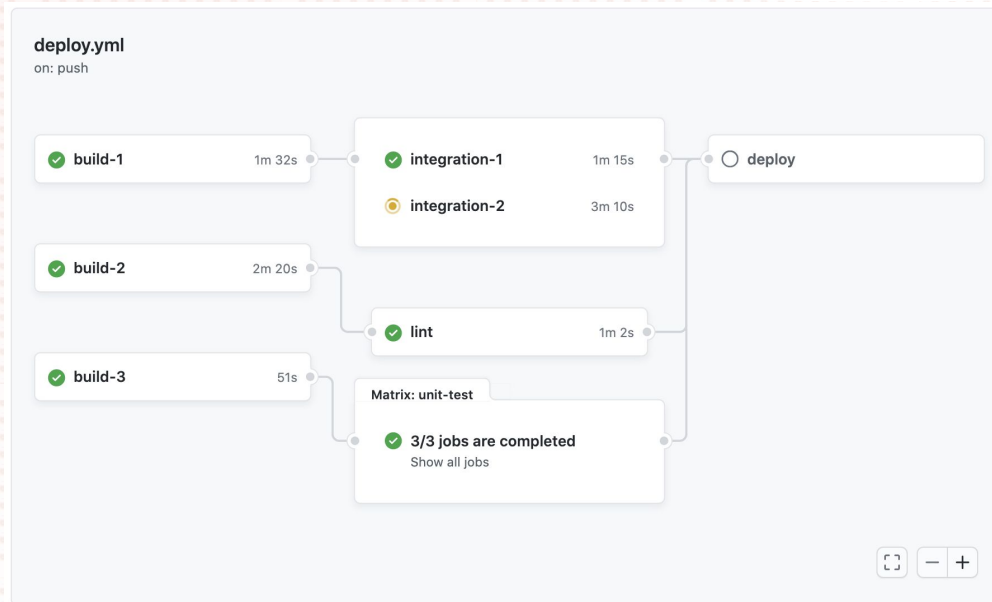


# PLATAFORMAS





# WORKFLOW



# YAML

```
in Terminal Help
... ! deploy.yml x js index.js
.github > workflows > ! deploy.yml
1  name: Push-to-EC2
2
3  on: push
4
5  jobs:
6    deploy:
7      name: Push to EC2 Instance
8      runs-on: ubuntu-latest
9
10     steps:
11       - name: Checkout the code
12         uses: actions/checkout@v1
13
14       - name: Deploy to my EC2 instance
15         uses: easingthemes/ssh-deploy@v2.1.5
16         env:
17           SSH_PRIVATE_KEY: "${ secrets.EC2_SSH_KEY }"
18           SOURCE: " ./"
19           REMOTE_HOST: "ec2-54-67-60-224.us-west-1.compute.amazonaws.com"
20           REMOTE_USER: "ubuntu"
21           TARGET: "/home/ubuntu/REST"
22
23       - name: Rerun the application
24         run: npm install
25         run: pm2 stop index
26         run: pm2 save
27         run: pm2 start index
28         run: pm2 save
```

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

## Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow template to get started.

Skip this and [set up a workflow yourself](#) →

### Workflows made for your repository Suggested

#### Simple workflow

By GitHub Actions

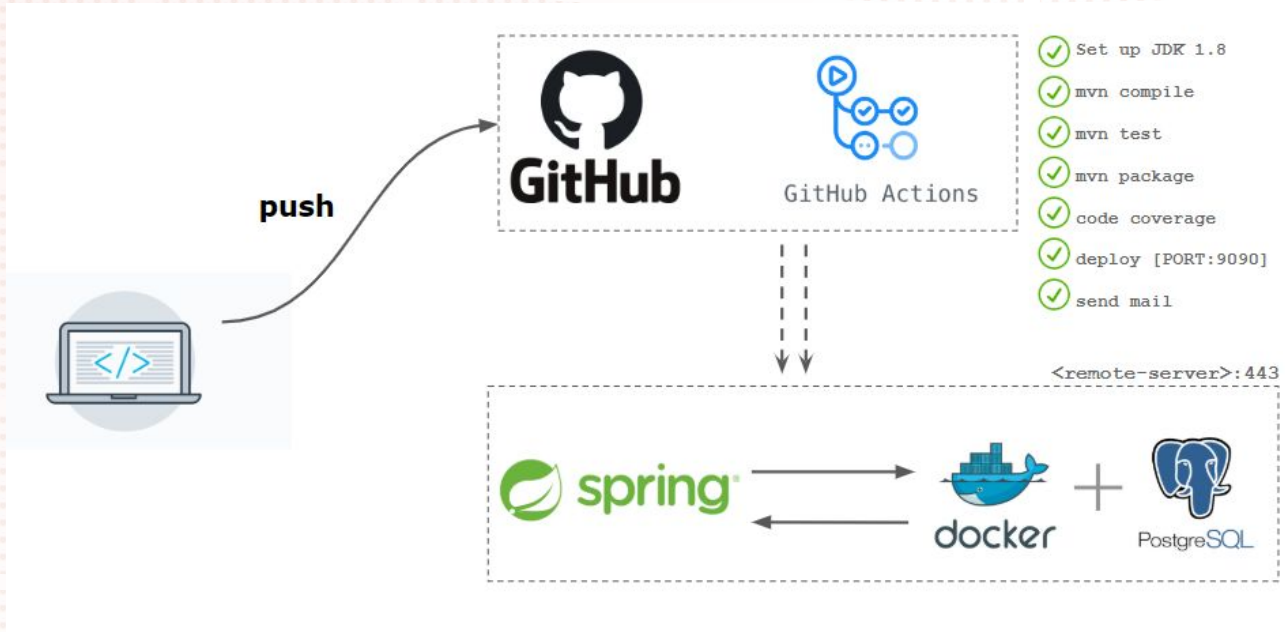
Start with a file with the minimum necessary structure.

Set up this workflow

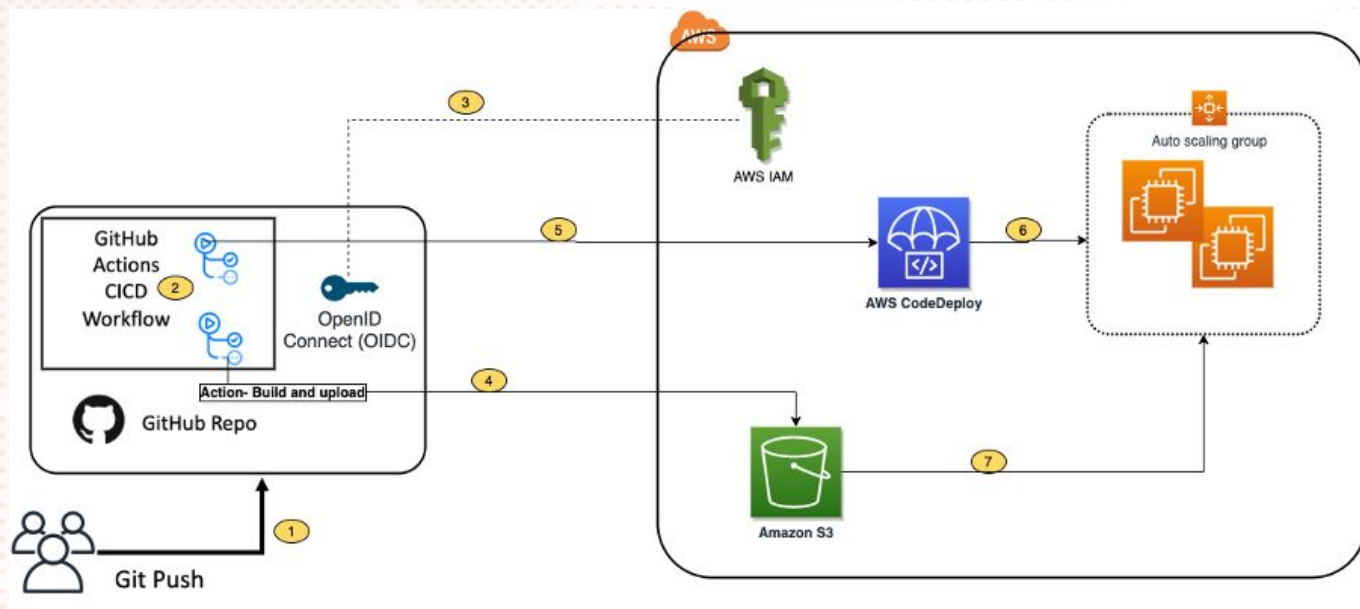
```
echo Hello, world!  
echo Add other actions to build,  
echo test, and deploy your project.
```

actions/starter-workflows

## Caso de uso



## Caso de uso 2





## Github in Actions



# REFERENCES

- Fowler, M. (2011). Continuous Integration. [Blog post]. Retrieved from <https://martinfowler.com/articles/continuousIntegration.html>
- Newman, D. (2016). Jenkins: The Definitive Guide. O'Reilly Media.
- Hashimoto, M., & Jenkins, C. (2016). Continuous Delivery with Docker and Jenkins. O'Reilly Media.
- GitHub. (2021). GitHub Actions Documentation. Retrieved from <https://docs.github.com/en/actions>
- Petazzoni, M. (2020). Using GitHub Actions for Continuous Integration and Continuous Deployment. [Blog post]. Retrieved from <https://container-solutions.com/using-github-actions-for-continuous-integration-and-continuous-deployment/>
- Poyraz, S. (2021). GitHub Actions for CI/CD. [Blog post]. Retrieved from <https://towardsdatascience.com/github-actions-for-ci-cd-8c406f060b07>