Desplegando aplicaciones en VM con GitHub Actions

LAUTARO CARRO

Contexto

- Utilizamos GitHub para los repositorios de código de nuestras aplicaciones.
- Tanto de manera manual o automática, a la hora de querer desplegar nuestra aplicación en una maquina virtual solemos pensar en FTP o SSH.

Objetivos

- Aprovechar las herramientas de GitHub Actions e implementar practicas de DevOps
- Evitar utilizar claves secretas, usuarios o contraseñas (FTP/SSH) de nuestro servidor en la cuenta de GitHub
- Simplificar nuestros workflows de CI/CD

@lauchacarro

Lautarocarro.blog

Lautaro Carro

FUNDADOR DE LATINO .NET ONLINE

DESARROLLADOR Y COUCH EN ALGEIBA

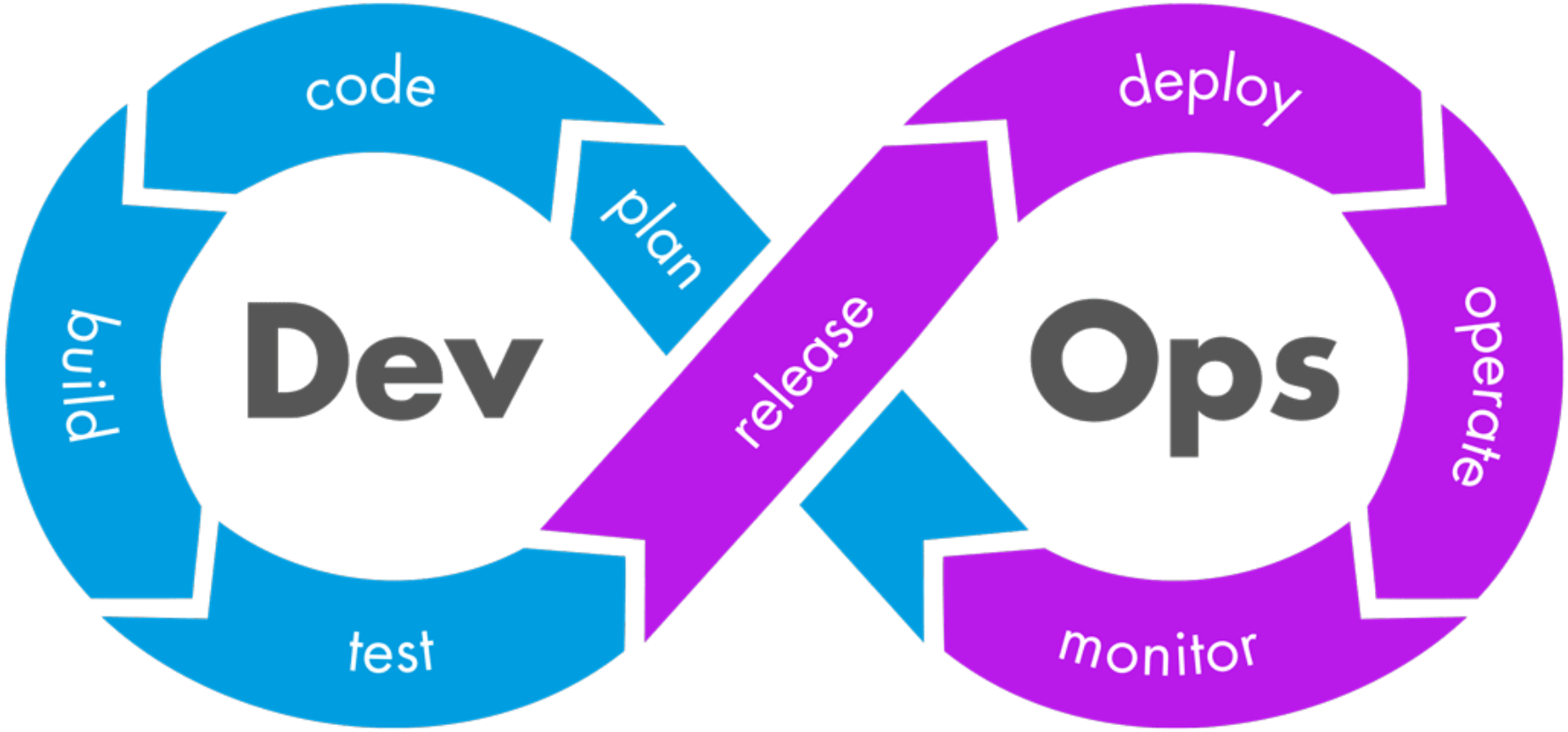
algeiba

Latino .NET Online

Agenda

- DevOps CI/CD
- GitHub Actions
  - Workflows
  - Jobs
  - Runners
  - Artifacts
- Build Job (Demo)
- Azure VM como Self-Hosted Runner
- Release Job (Demo)

DevOps


code
plan
build
Dev
test
release
deploy
operate
Ops
monitor

DevOps: CI/CD

- La **CI/CD** es un método para distribuir las aplicaciones mediante el uso de la **automatización** en las etapas del desarrollo de aplicaciones.

CI/CD
¡Para Todos!

```yaml
name: dotnet package

on: [push]

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v3
      - name: Setup .NET  SDK
        uses: actions/setup-dotnet@v2
        with:
          dotnet-version:  6.x
      - name: Install dependencies
        run: dotnet restore
      - name: Build
        run: dotnet build --configuration Rel
      - name: Test
        run: dotnet test --no-restore --verbos
```

GH Actions Workflows

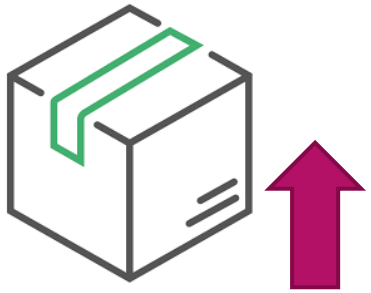
Build Job: Publicar el Artefacto

Integración y entrega continua:

Compilamos el código

Corremos los unit tests

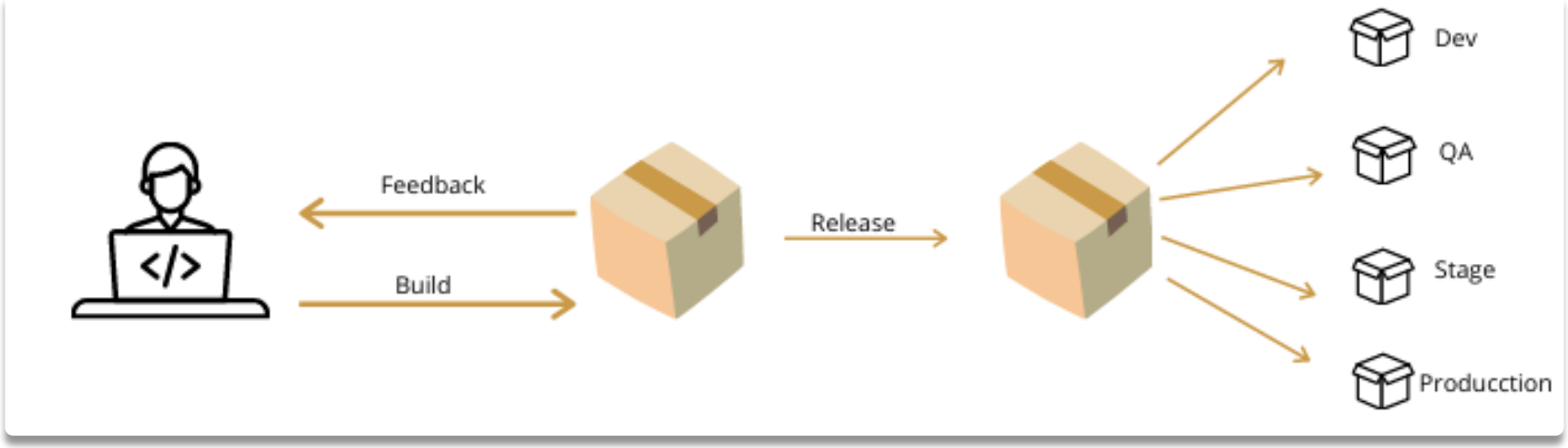
(Artefacto)



Publicamos el paquete


LIVE DEMO

Build y Release por separado

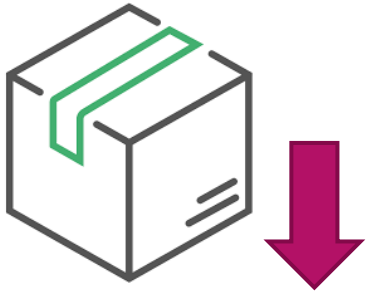


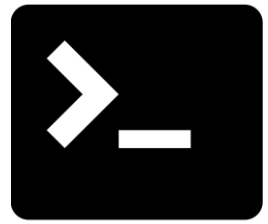
- Separating Build and Release Pipelines for Effective DevOps | by Nishu_Dissanayake | Bits and Pieces (bitsrc.io)

Release Job: Descargar el Artefacto

Implementación continua:

(Artefacto)





Descargamos el paquete

Guardamos los archivos

Ejecutamos el servicio

Self-Hosted Runners

Build

GitHub's Runner

Release

Self-hosted runners

Un **Runner** es un servidor que ejecuta tus flujos de trabajo cuando se activan. Cada Runner puede ejecutar un job individual a la vez.


LIVE DEMO

Conclusiones

- Creamos un workflow de CI/CD con Github Actions
- Creamos nuestro Self-Hosted Runner para desplegar nuestra aplicación directamente en el servidor
- Separamos nuestro workflow en Build y Release para una mejor implementación de la metodología de DevOps

Próximos Pasos:

- Publicación de imágenes Docker en el Release Job
- Utilizar Download workflow artifact para descargar Artefactos anteriores
- Reutilizar Workflows en GitHub Actions

@lauchacarro

Lautarocarro.blog

Muchas Gracias

DESPLEGANDO APLICACIONES EN UNA MAQUINA VIRTUAL CON GITHUB ACTIONS

algeiba

Latino .NET Online

Referencias

- **Deploy .NET 6 Web App With GitHub Actions To Self-Hosted Machine (amelspahic.com)**
- Instalación de .NET en Ubuntu - .NET | Microsoft Docs
- Separating Build and Release Pipelines for Effective DevOps | by Nishu_Dissanayake | Bits and Pieces (bitsrc.io)
- ¿Qué son la integración/distribución continuas (CI/CD)? (redhat.com)