



#GlobalAzure

#GABMUGPeru

#GABMUGPeru

Uso de plantillas pipelines YAML en Azure DevOps

Ricardo Barreno



#GlobalAzure

#GABMUGPeru



Speaker

Ricardo Barreno
Azure Solutions Architect



#GlobalAzure

Agenda

- Azure DevOps
- Azure Pipelines
- Ejemplo

#GABMUGPeru

Azure DevOps



#GlobalAzure

Azure DevOps

- Plataforma integral de desarrollo colaborativo de software de Microsoft.
- Automatización de procesos de entrega continua.
- Gestión completa del ciclo de vida del desarrollo de Software.
- Colaboración mejorada entre equipos de desarrollo y operaciones.
- Seguimiento y gestión eficaz de tareas y problemas.



Azure DevOps - Servicios



Azure Boards

Plan, track, and discuss work across teams, deliver value to your users faster.



Azure Repos

Unlimited cloud-hosted private Git repos. Collaborative pull requests, advanced file management, and more.



Azure Pipelines

CI/CD that Works with any language, platform, and cloud. Connect to GitHub or any Git provider and deploy continuously to any cloud.



Azure Test Plans

The test management and exploratory testing toolkit that lets you ship with confidence.



Azure Artifacts

Create, host, and share packages. Easily add artifacts to CI/CD pipelines.

#GABMUGPeru

Azure Pipelines



#GlobalAzure

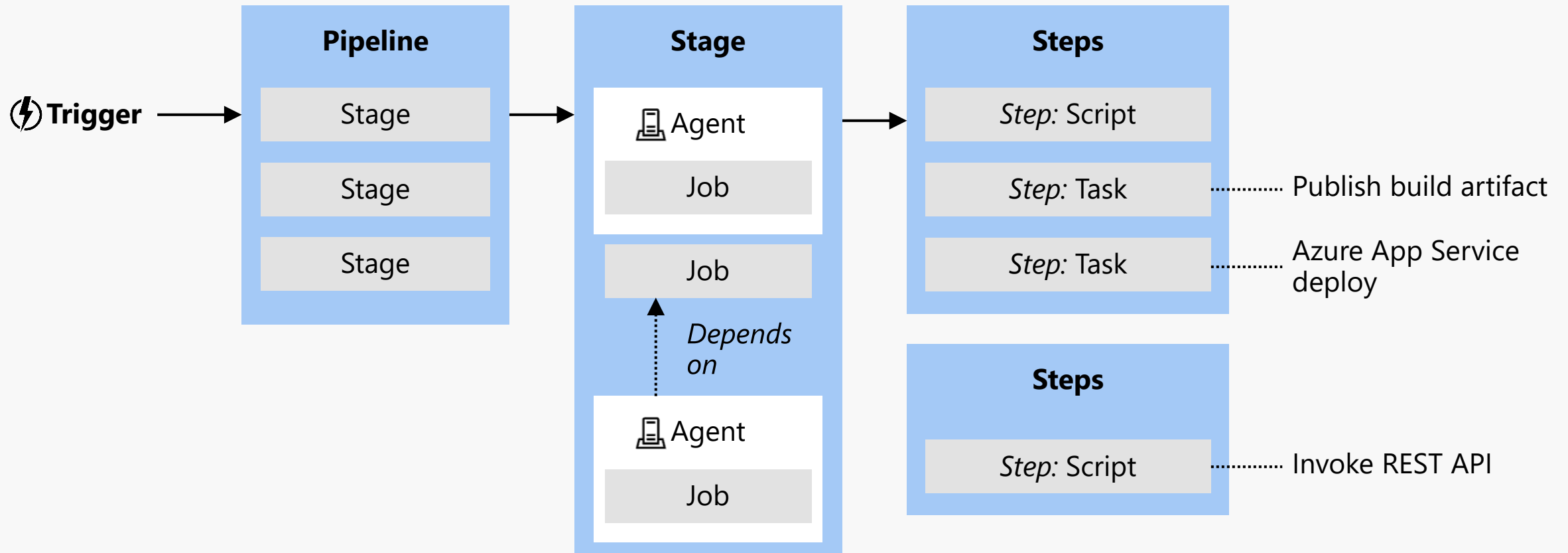
Azure Pipelines

- Servicio de CI/CD en la nube que permite compilar, probar y desplegar aplicaciones.
- Compatibilidad con cualquier lenguaje, plataforma o nube.
- Escalabilidad para adaptarse a proyectos pequeños y grandes.
- Flexibilidad para definir flujos de trabajo personalizados.



Azure Pipelines

Azure Pipelines – Key Concepts



Azure Pipelines - YAML

- Lenguaje de marcado utilizado para definir pipelines en Azure DevOps.
- Simplificación de la configuración y mantenimiento de pipelines.
- Legibilidad mejorada y sintaxis clara para definir pipelines.
- Facilidad de control de versiones al ser archivos de texto plano.
- Flexibilidad para adaptar flujos de trabajo a diferentes necesidades.



Azure Pipelines

Azure Pipelines – Plantillas YAML

- Fragmentos reutilizables de código YAML que permiten definir configuraciones de pipelines.
- Simplificación y estandarización de la configuración del pipeline.
- Reutilización de patrones comunes de configuración en múltiples pipelines.
- Facilita el mantenimiento al centralizar la lógica de configuración en un solo lugar.



Azure Pipelines

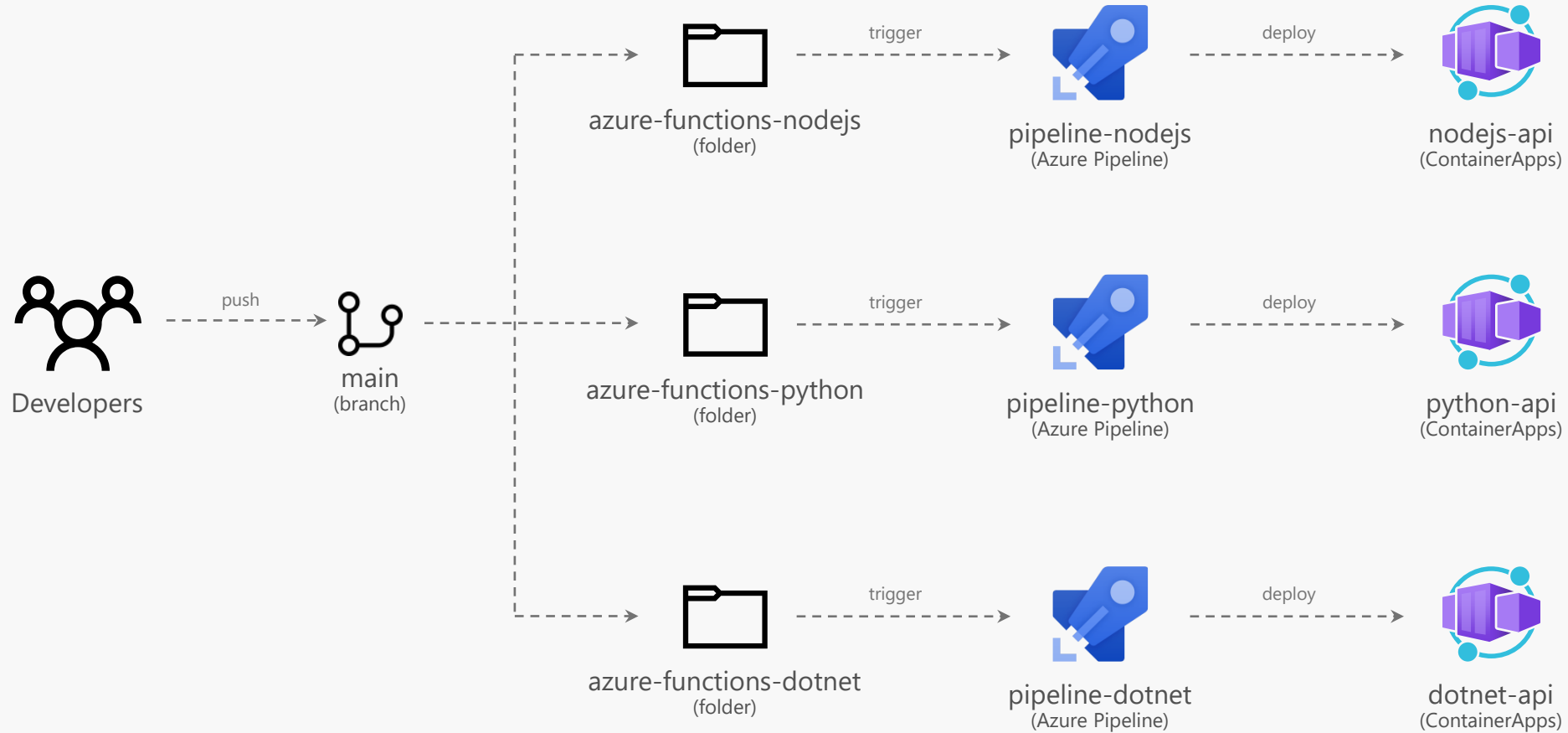
#GABMUGPeru

Ejemplo



#GlobalAzure

Ejemplo – Diagrama



Ejemplo – Plantillas YAML

```
name: dotnet-pipeline

trigger:
  branches:
    include:
      - main
      - develop
  paths:
    include:
      - src/azure-function-dotnet

pool:
  vmImage: 'ubuntu-latest'
```

```
extends:
  template: templates/base-pipeline.yml
  parameters:
    language: dotnet
    workingDirectory: src/azure-function-dotnet
    repositoryName: azure-function-dotnet
```

dotnet-pipeline.yml

```
parameters:
  - name: language
    type: string
    values:
      - dotnet
      - python
      - nodejs
  - name: workingDirectory
    type: string
  - name: repositoryName
    type: string
```

```
variables:
  - template: vars.yml
    parameters:
      language: ${ parameters.language }
```

```
stages:
  - ${ if eq(parameters.language, 'dotnet') }:
    - template: dotnet-build-stage.yml
      parameters:
        workingDirectory: ${ parameters.workingDirectory }

  - template: containerization-stage.yml
    parameters:
      containerRegistry: '${ variables.containerRegistry }'
      repository: '${ variables.dockerHubUsername }/${ parameters.repositoryName }'
      buildContext: ${ parameters.workingDirectory }

  - template: deployment-stage.yml
    parameters:
      azureSubscription: '${ variables.azureSubscription }'
      containerAppName: '${ variables.containerAppName }'
      resourceGroup: '${ variables.resourceGroupName }'
      imageToDeploy: '${ variables.containerRegistryServer }/${ parameters.repositoryName }:$(Build.BuildId)'
```

templates/base-pipeline.yml

Ejemplo – Plantillas YAML

```
parameters:
- name: language
  type: string
  values:
    - dotnet
    - python
    - nodejs

variables:
- name: dockerHubUsername
  value: richibarreno
- name: containerRegistry
  value: DockerHub
- name: containerRegistryServer
  value: docker.io
- name: azureSubscription
  value: GlobalAzure-Template-Pipelines
- name: containerAppName
  ${if eq(parameters.language, 'dotnet')}:
    value: dotnet-containerapp
  ${if eq(parameters.language, 'python')}:
    value: python-containerapp
  ${if eq(parameters.language, 'nodejs')}:
    value: nodejs-containerapp
- name: resourceGroupName
  value: GlobalAzure-Template-Pipelines
```

template/vars.yml

```
parameters:
- name: workingDirectory
  type: string

stages:
- stage: build
  displayName: 'Build'

  jobs:
  - job: building
    displayName: 'Building'

    steps:
    - task: DotNetCoreCLI@2
      name: restore
      displayName: 'Restore packages'
      inputs:
        command: 'restore'
        feedsToUse: 'select'
        projects: '**/*.csproj'
        workingDirectory: ${parameters.workingDirectory}

    - task: DotNetCoreCLI@2
      name: build
      displayName: 'Build app'
      inputs:
        command: 'build'
        projects: '**/*.csproj'
        workingDirectory: ${parameters.workingDirectory}
```

template/dotnet-build-stage.yml

Ejemplo – Plantillas YAML

```
parameters:
- name: language
  type: string
  values:
    - dotnet
    - python
    - nodejs

variables:
- name: dockerHubUsername
  value: richibarreno
- name: containerRegistry
  value: DockerHub
- name: containerRegistryServer
  value: docker.io
- name: azureSubscription
  value: GlobalAzure-Template-Pipelines
- name: containerAppName
  ${{ if eq(parameters.language, 'dotnet') }}:
    value: dotnet-containerapp
  ${{ if eq(parameters.language, 'python') }}:
    value: python-containerapp
  ${{ if eq(parameters.language, 'nodejs') }}:
    value: nodejs-containerapp
- name: resourceGroupName
  value: GlobalAzure-Template-Pipelines
```

template/vars.yml

```
parameters:
- name: containerRegistry
  type: string
- name: repository
  type: string
- name: buildContext
  type: string

stages:
- stage: containerization
  displayName: 'Containerization'

  jobs:
    - job: containerizing
      displayName: 'Containerizing'

      steps:
        - task: Docker@2
          name: build_image
          displayName: 'Build image'
          inputs:
            containerRegistry: '${{ parameters.containerRegistry }}'
            repository: '${{ parameters.repository }}'
            command: 'build'
            Dockerfile: '${{ parameters.buildContext }}/Dockerfile'
            buildContext: '${{ parameters.buildContext }}'

        - task: Docker@2
          name: push_image
          displayName: 'Push image'
          inputs:
            containerRegistry: '${{ parameters.containerRegistry }}'
            repository: '${{ parameters.repository }}'
            command: 'push'
```

template/containerization-stage.yml

Ejemplo – Plantillas YAML

```
parameters:
- name: language
  type: string
  values:
    - dotnet
    - python
    - nodejs

variables:
- name: dockerHubUsername
  value: richibarreno
- name: containerRegistry
  value: DockerHub
- name: containerRegistryServer
  value: docker.io
- name: azureSubscription
  value: GlobalAzure-Template-Pipelines
- name: containerAppName
  ${if eq(parameters.language, 'dotnet')}:
    value: dotnet-containerapp
  ${if eq(parameters.language, 'python')}:
    value: python-containerapp
  ${if eq(parameters.language, 'nodejs')}:
    value: nodejs-containerapp
- name: resourceGroupName
  value: GlobalAzure-Template-Pipelines
```

template/vars.yml

```
parameters:
- name: azureSubscription
  type: string
- name: containerAppName
  type: string
- name: resourceGroup
  type: string
- name: imageToDeploy
  type: string

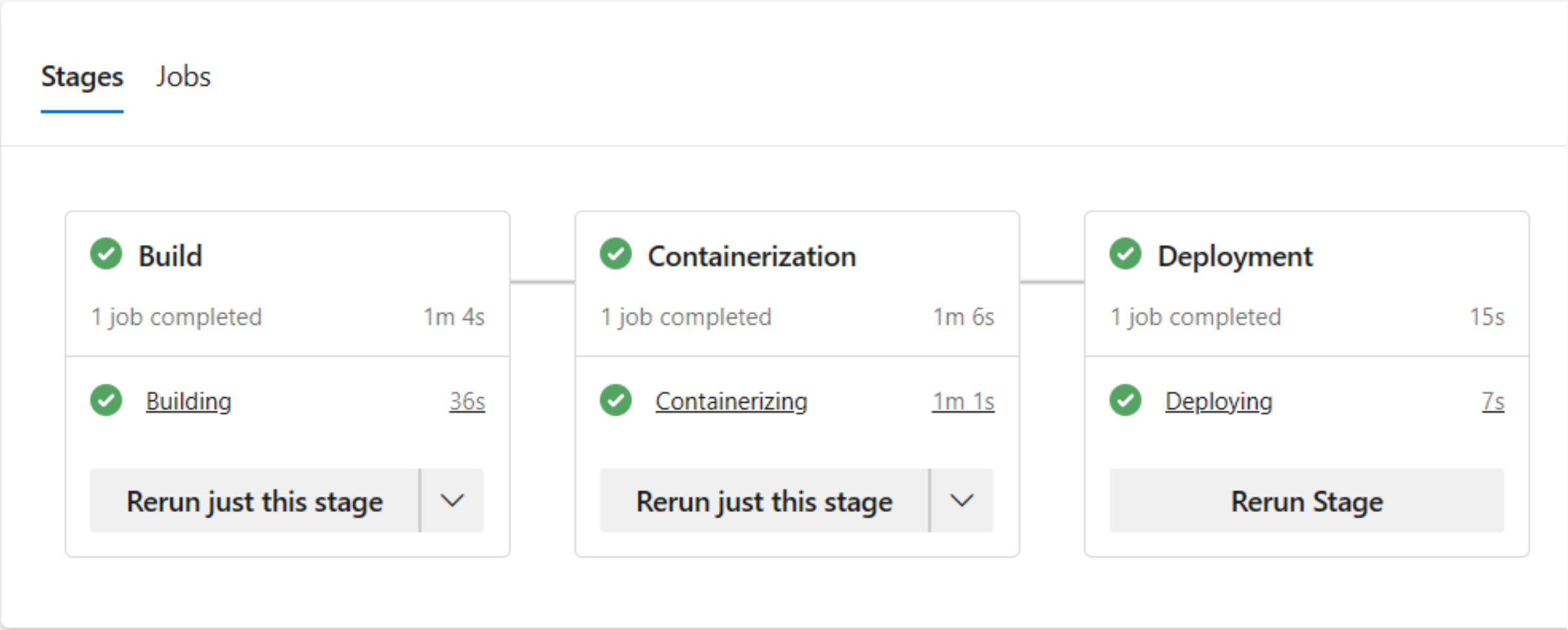
stages:
- stage: deployment
  displayName: 'Deployment'

  jobs:
    - job: deploying
      displayName: 'Deploying'

      steps:
        - task: AzureContainerApps@1
          name: deploy_containerapp
          displayName: 'Deploy ContainerApp'
          inputs:
            azureSubscription: '${parameters.azureSubscription}'
            containerAppName: '${parameters.containerAppName}'
            resourceGroup: '${parameters.resourceGroup}'
            imageToDeploy: '${parameters.imageToDeploy}'
```

template/deployment-stage.yml

Ejemplo – Ejecución pipeline



Ejemplo - Repositories

[AzureDevOps-Pipeline-Template \(github.com\)](#)

[AzureDevOps-Pipeline-Template \(Azure DevOps\)](#)

Patrocinadores



Microsoft



25 mibanco

años



Colabora



nuvem

