

# Azure DevOps

miércoles, 28 de junio de 2023 12:56

## DevOps:

Development & Operations

Desarrollo de la aplicación y despliegue de la misma

Cultura que converge entre las Herramientas & Personas & Procesos

Cultura de mejora continua

Automatización de procesos

No es posible aplicar agilidad sin DevOps!

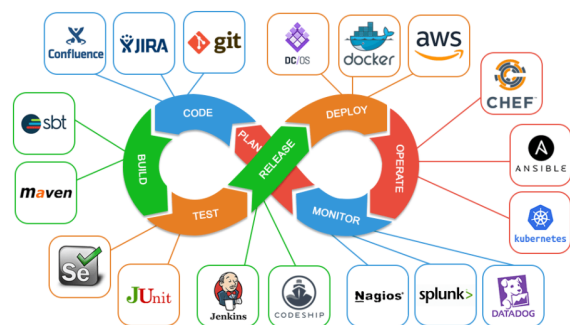
DevOps != Una metodología agil

DevOps != CI/CD pero si va de la mano con este proceso

Ciclo de vida del software desde DevOps

## Azure DevOps:

- Conjunto de herramientas que acompañan el ciclo de vida del software usando la cultura DevOps / Suite DevOps de MS.
- Cubre todo el ciclo de vida del software
- 



Para ver más acerca de DevOps:

- <https://learn.microsoft.com/es-mx/training/paths/evolve-your-devops-practices/>
- <https://www.amazon.com/DevOps-Modern-Enterprise-Practices-Organizations/dp/1942788193>

Organización: Funciona como un agrupador de proyectos

Tipos de board y características:

- Basic: Board limitado, muy parecido a Trello
- CMMI: Aplicar metodología CMMI, proceso un poco más complicado que el agilidad (no se usa mucho)
- Agile: Orientado a la aplicación de metodologías ágiles como Kanban o SCRUM
- SCRUM: Especializado en la aplicación de SCRUM.

Secciones de DevOps:

- Overview: Información general del proyecto, la sección 'Dashboards' me permite crear visualizaciones basadas en diferentes factores como releases, PR's, commits, y demas.
- Wiki: Documentación del proyecto.
- Boards / Work Item: Cualquier actividad que se deba de hacer en el proyecto.
- Repos: Repositorios
- Pipelines: Integración.
- Artifacts: Creación de librerías propias.

Para poder agregar a un usuario externo a un proyecto, primero deberá de ser agregado a la organización

Azure Active Directory: Servicio de Azure orientado a grandes empresas con muchos empleados que necesitan más controles de seguridad, además, de una estructura jerárquica bien definida.

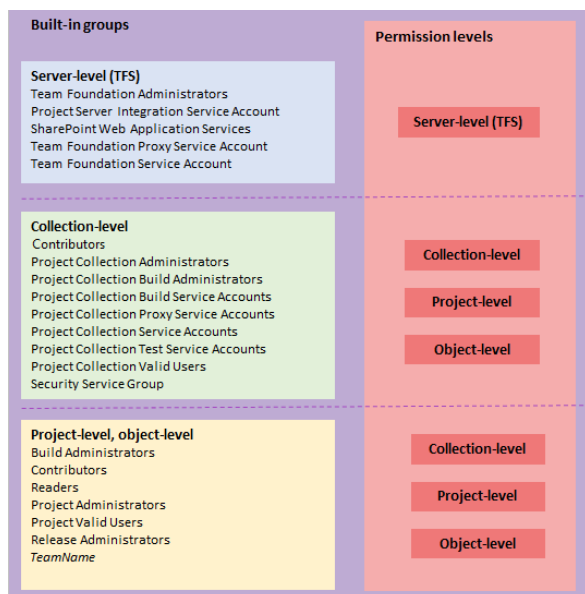
Configuraciones básicas de Seguridad:

Configuraciones de la organización, Permisos.

Se pueden crear grupos, unir personas para que tengan ciertas configuraciones de seguridad.

**Membership management**

**Permission management**



Backlog: Grupo de tickets (tareas) que están pendientes por atenderse.

Uso de Tags en los tickets para referirse a que modulo hace parte el mismo.

Criterios de aceptación: Puntos a considerar para definir si el ticket se completó o no.

Pull request: Petición para que el código pase de rama en rama.

Se puede asociar una rama creada con uno de los items del backlog.

Pipeline CD en azure DevOps:

Pipeline: Secuencia de comandos que realizan una ejecución de la aplicación para comprobar que este compilando, funcionando y corriendo las pruebas unitarias.

Azure repos Git: Azure lee los repositorios que se encuentran dentro del proyecto.

Estructura de YAML

trigger:

- Master (rama en la que se va a fijar para empaquetar el código)

Pool:

- Ubuntu-latest (creación de máquina virtual con la última versión estable de ubuntu)

Steps:

- Task: nombre de la tarea a ejecutar (azure define el nombre dependiendo de la tarea)

Inputs:

- o VersionSpec: 'versión de lenguaje de programación que va a instalar'

DisplayName: "nombre de la tarea a ejecutar"

- Script: | (script para correr en la VM ubuntu)

Npm install (instalar todas las dependencias del proyecto)

Npm run build (compilar)

DisplayName: "nombre de la tarea a ejecutar"

Código utilizado para la inicialización del pipeline

- script: |

npm install

displayName: 'npm install'

- script: |

npm run build

displayName: 'npm run build'

- task: CopyFiles@2

inputs:

Contents: 'build/\*\*' (copiar todos los archivos que se encuentren dentro de la carpeta build)

TargetFolder: '\$(build.ArtifactStagingDirectory)' (copiar a carpeta interna de staging del agente todo los contenidos de la carpeta build)

- task: ArchiveFiles@2 (comprimir a un .zip todos los archivos que se compilaron en la build)

inputs:

rootFolderOrFile: '\$(Build.ArtifactStagingDirectory)'

```
includeRootFolder: true
archiveType: zip (tipo de archivo resultante)
archiveFile: '${Build.ArtifactStagingDirectory}/${Build.BuildId}.zip' (nombre del archivo resultante)
replaceExistingArchive: true

- task: PublishBuildArtifacts@1 (publicar el archivo .zip para que otras secciones de DevOps puedan disponibilizar el .zip)
inputs:
  PathToPublish: '${Build.ArtifactStagingDirectory}'
  ArtifactName: 'drop'
```

Continuos deployment: Cada vez que exista / se genere un nuevo paquete va a hacer todo el flujo para poder desplegarlo.