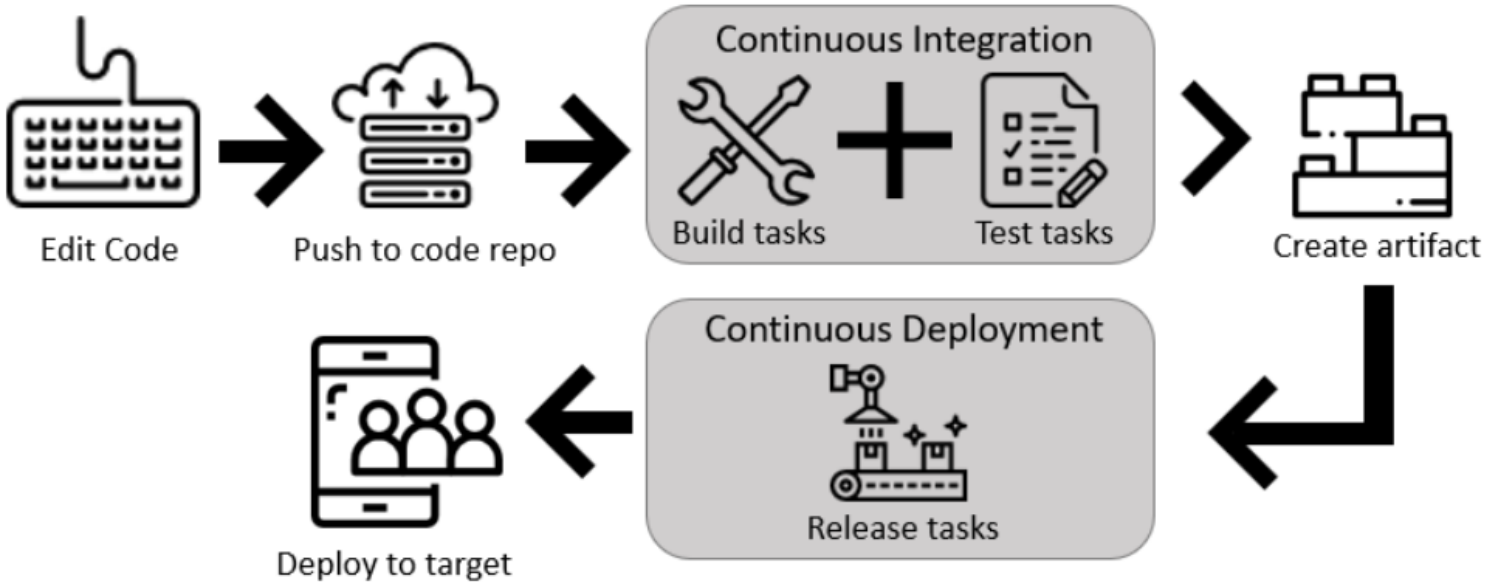
YAML

Define pipelines using the Classic interface

Create and configure pipelines in the Azure DevOps web portal with the Classic user interface editor. You define a *build pipeline* to build and test your code, and then to publish artifacts. You also define a *release pipeline* to consume and deploy those artifacts to deployment targets.



1.Configure Azure Pipelines to use your Git repo.

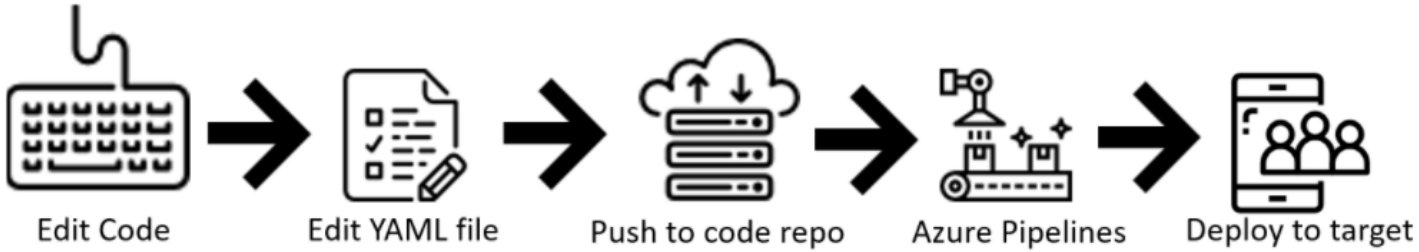
2.Use the Azure Pipelines classic editor to create and configure your build and release pipelines.

3.Push your code to your version control repository. This action triggers your pipeline and runs tasks such as building or testing code.

The build creates an artifact that's used by the rest of your pipeline to run tasks such as deploying to staging or production.
Your code is now updated, built, tested, and packaged. It can be deployed to any target.

Define pipelines using YAML syntax

Define your pipeline in a YAML file called azure-pipelines.yml with the rest of your app.



- The pipeline is versioned with your code. It follows the same branching structure. You get validation of your changes through code reviews in pull requests and branch build policies.
- Every branch you use can modify the build policy by modifying the azure-pipelines.yml file.
- A change to the build process might cause a break or result in an unexpected outcome. Because the change is in version control with the rest of your codebase, you can more easily identify the issue.

1. Configure Azure Pipelines to use your Git repo.
2. Edit your azure-pipelines.yml file to define your build.
3. Push your code to your version control repository. This action kicks off the default trigger to build and deploy and then monitor the results.

Pipelines as Code with YAML

Why you should use YAML

There are a few advantages to using YAML files:

1.Easily readable by humans, YAML files are expressive and extensible.
2.They are easy to implement and use.
3.They are easily portable between programming languages.
4.They match the native data structures of agile languages.
5.YAML files have a consistent model to support generic tools.
6.They support one-pass processing.
7.Convenient to use, so you no longer need to add all of our parameters to the command line.
8.Maintenance can be done – YAML files can be added to the source control to track the changes.
9.Flexibility – You can create much complex structures using YAML than you can use on command line

YAML Basics

- YAML is a data serialization language which means you can represent data in the format of YAML.
- Each name value pair in YAML is represented as <name>: <value>
- Representing data in the form of name value pairs
- Both Machine readable & human readable
- Uses significant Whitespaces

- YAML is a data serialization language which means you can represent data in the format of YAML.
- Each name value pair in YAML is represented as <name>: <value>

- If we think of any data the values are
  1. singular or simple:
     - you can represent text
     - you can represent numbers
     - you can represent Booleans (True, False, yes, no)

     - **Simple Text notations**

     name: DevOps Forum
     name: 'DevOps Forum'
     name: "DevOps Forum"

- **Multi line text notations**

```
about-us: | This forum was started by Group of people
about-us: > forum to discuss about DevOps architecture
```

- **Number notations**

```
Number of members: 10
```

- **Boolean notations**

```
is_online: yes
is_online: true
```

2.plural or list: can be represented with significant whitespace or in the form of **[]**. Each item in normal notation starts with – (list item)Example with standard notation

```
Topics:
  - Azure DevOps
  - Git
  - Docker
  - Kubernetes
```

- In array notation

```
Topics: [“Azure DevOps", “Git", “Docker", “Kubernetes“]
```

3.complex or object: An object has multiple name & value pairs

- Sample address

```
address:
  blockno: 123456
  building: opera
  area: Gachibowli
  landmark: DLF
  city: Hyderabad
```

- The whole yaml is generally created with .yml or .yaml extensions
- Lets create a simple file devopsforum.yml

```
name: DevOps Forum
about-us: This forum was started by Group of people
Number of members: 10
is_online: true
Topics:
  - Azure DevOps
  - Git
  - Docker
  - Kubernetes
address:
  blockno: 123456
  building: opera
  area: Gachibowli
  landmark: DLF
  city: Hyderabad
```

After writing YAML validate with any online YAML validator such as http://www.yamllint.com/

Relationship to JSON and XML

XML	JSON	YAML

<Servers>
<Server>
<name>Server1</name>
<owner>John</owner>
<created>123456</created>
<status>active</status>
</Server>
</Servers>
	{
	Servers: [
	{
	name: Server1,
	owner: John,
	created: 123456,
	status: active
	}
	]
	}
		Servers:
		- name: Server1
		owner: John
		created: 123456
		status: active

YAML schema documentation

https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html

https://docs.microsoft.com/azure/devops/pipelines/yaml-schema