

<https://github.com/MicrosoftDocs/pipelines-dotnet-core.git> - git botnet project

https://github.com/prabhudev74/dotnet

botnet core pipeline steps

<https://docs.microsoft.com/en-us/azure/devops/pipelines/?view=azure-devops>

botnet framework pipeline steps

<https://docs.microsoft.com/en-us/azure/devops/pipelines/apps/aspnet/build-aspnet-4?view=azure-devops>

its azure service that lets you to build ,test and deploy applications on azure cloud and other supported platforms.

i)platform and language independent

ii)freedom to work with containers

iii)deployment to other cloud vendors

iv)support for open source

Azure Pipelines automatically builds and tests code projects to make them available to others. It works with just about any language or project type. Azure Pipelines combines continuous integration (CI) and continuous delivery (CD) to constantly and consistently test and build your code and ship it to any target.

2 types of pipelines

i) Yaml - stored in the form of YAML code. stored in repos. it has multiple stages such as build and release(both fill be performed in single stage)

we can deploy condition based pipelines.

ii)class c - graphical representation.here the build stage and release stages are performed separately.

| YAML | Classic |
| --- | --- |
| track ur changes , modifications | track history but can’t compare what exactly changed |
| condition based pipelines and dependency based pipelines |  |
| multi staging(build ,uat) |  |
| template | drag and drop graphical interface |
| present under azure repos |  |
| stored in repos particular branch | pre and post approvals |

language build tool

java ——— maven

.net ——— MS build

.net core —- .net core task

.net core —— .net

angular js —— NPN

here We have used MS build to compile, build and to generate artefacts of a source code.

**Agents**

agent is used to build to compile, build and to generate artefacts of a source code.

agent will be created on remote server

1)hosted agents — managed by micro soft. naming convention is azure pipeline

2)self hosted agent —- can install required softwares and perform trouble shooting steps

**Library** - stores environment variables

predefined variables — defined by micro soft

pipeline variables — variables used in within the pipeline

variables group— variables used in group of pipelines

**Task groups** - re usable tasks will be grouped together.

**deployment groups** - here the code is build and deployed there it self. mostly for vm’s.

**repos -**

files — project code developed by developer will be under files in repos.

commit - committed changes

push - changes that are pushes

branches - diff. types of branches

tags - recognition , for particular release all the features will be stored

pull request -to validate code from one branch to another brach pull request approve first it will be done by i)sonar qube review

then reviewer will review ii)reviewer review

Gated build — after raising a PR to pass code from feature to dev then gated build will run. here we do quality check.

continuous integration — after the PR completed CI will run. Artefacts will generate.

while creating the pipeline, artefacts will be created. these artefacts are used to deploy the code. Along with artefacts , stages are also created.

scheduled build

**Creating pipelines**

pipelines will be created in dev , hot fix , release branches mostly

$(variable name)

\*\*\*.\*\sln —- to select all sln files

restore - references will restore

build - it will build and compile solutions

publish - to generate zip packages

publish artefact - to store the zip file to drop location