**International Journal of Engineering, Science & Mathematics**
Vol. 12 Issue 01, January 2023,
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

# Creating a Continuous Integration for .Net Using Azure DevOps

Srilatha Boga
Principal Full Stack Developer
Working at Visionworks, Iselin, New Jersey, United States.
Email: Srilatha0515@gmail.com

A Continuous pipeline is one of the best practices software development and it is a series of steps that must be performed in order to deliver a new version of software.

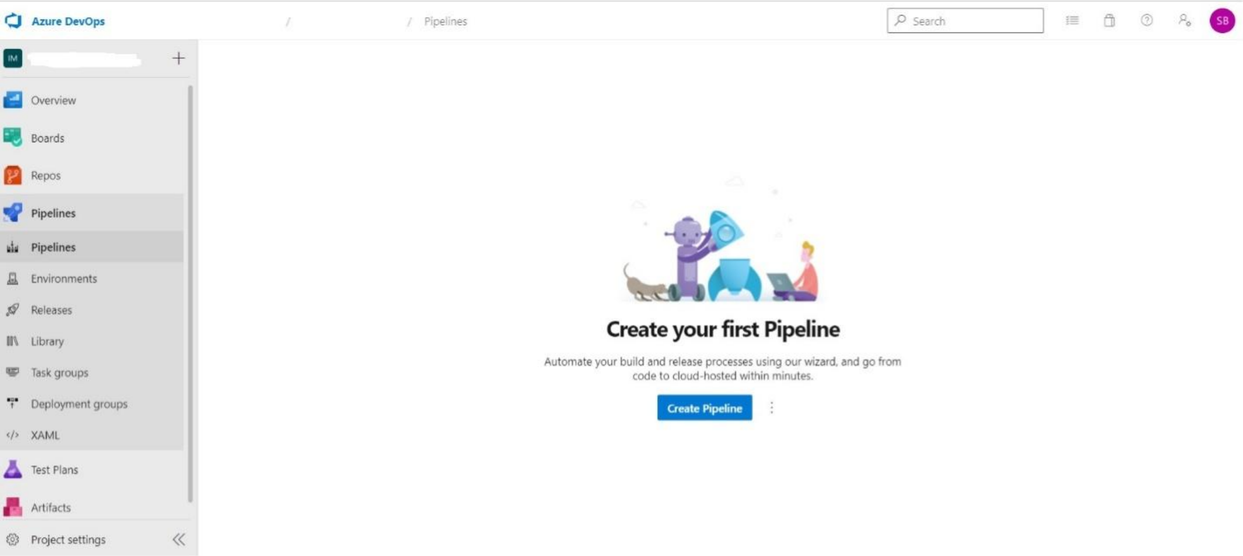
Developers practicing continuous integration merge their changes back to the main branch as often as possible. The developer's changes are validated by creating a build and running automated tests against the build. By doing so, you avoid integration challenges that can happen when waiting for release day to merge changes into the release branch.

Continuous integration puts a great emphasis on testing automation to check that the application is not broken whenever new commits are integrated into the main branch.

Step 1: Login to Azure DevOps account.

Step 2: Click on Pipelines from the left side nav.

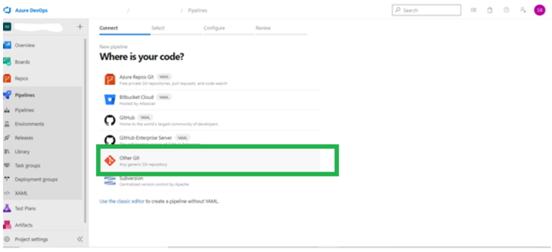
Step 3: Click on "Create Pipeline" button to create a new pipeline.



Create your first Pipeline

Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes.

Step 4: Under "Where is your code?" option, select "Other Git"



Where is your code?

Azure Repos Git — Free private Git repositories, pull requests, and code search

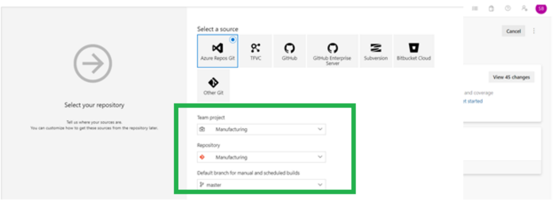
Bitbucket Cloud — Hosted by Atlassian

GitHub — Home to the world's largest community of developers

Other Git — Any generic Git repository

Subversion — Centralized version control by Apache

Use the classic editor to create a pipeline without YAML.



Select your repository

Tell us where your sources are.
You can customize how to get these sources from the repository later.

Team project: Manufacturing

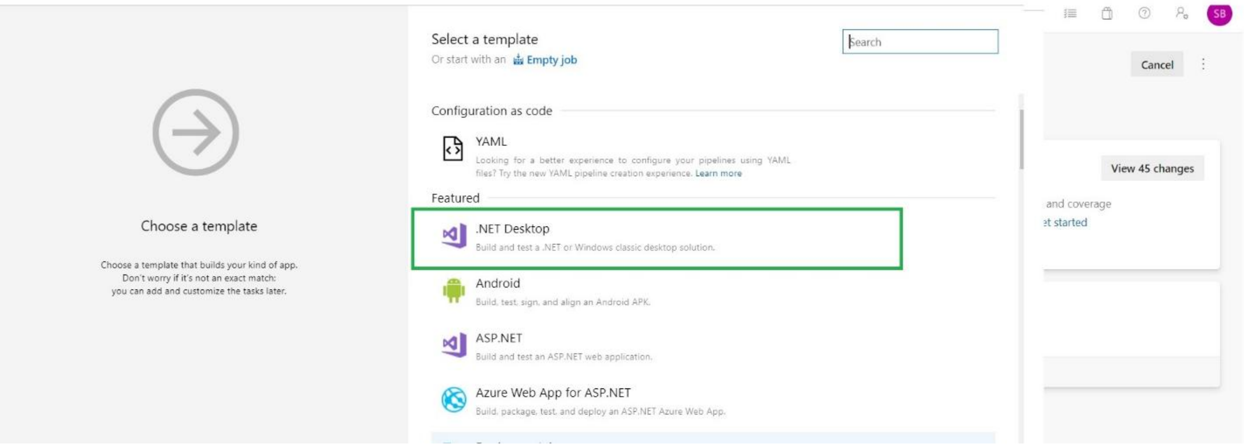
Repository: Manufacturing

Default branch for manual and scheduled builds: master

Step 5: Next, select a repository type from "Select a Source" options like Azure Repos Git/Other Git.

Here I have choose "Azure Repos Git" option. Since the repo is in the same Azure, select the project from the dropdown otherwise provide connection details to external Git service connection.

Step 6: Select a template type. Here it i have choosen ".Net Desktop"



Choose a template

Choose a template that builds your kind of app.
Don't worry if it's not an exact match:
you can add and customize the tasks later.

Select a template
Or start with an Empty job

Configuration as code

YAML
Looking for a better experience to configure your pipelines using YAML files? Try the new YAML pipeline creation experience. Learn more

Featured

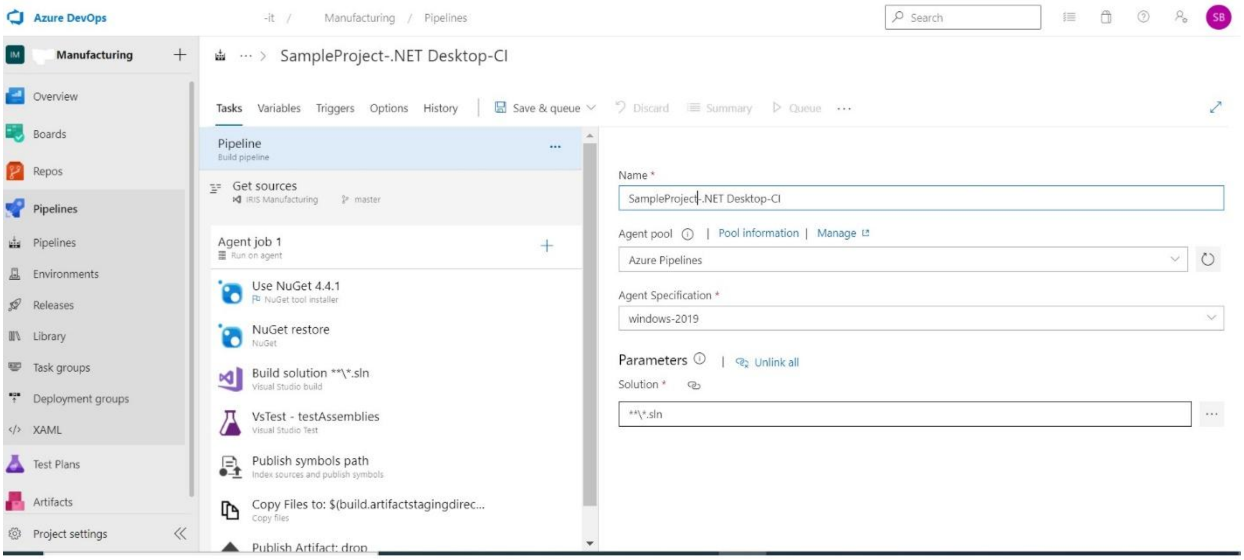
.NET Desktop
Build and test a .NET or Windows classic desktop solution.

Android
Build, test, sign, and align an Android APK.

ASP.NET
Build and test an ASP.NET web application.

Azure Web App for ASP.NET
Build, package, test, and deploy an ASP.NET Azure Web App.

Step 7: A template has been created. This has five configuration steps such as Tasks, Variables, Triggers, Options and History. Click on "Save and queue", then provide the comment details and then click on "Save and run".



SampleProject-.NET Desktop-CI

Pipeline
Build pipeline

Get sources
IRIS Manufacturing master

Agent job 1
Run on agent

Use NuGet 4.4.1
NuGet tool installer

NuGet restore
NuGet

Build solution ***.sln
Visual Studio build

VsTest - testAssemblies
Visual Studio Test

Publish symbols path
Index sources and publish symbols

Copy Files to: $(build.artifactstagingdirec...
Copy files

Publish Artifact: drop

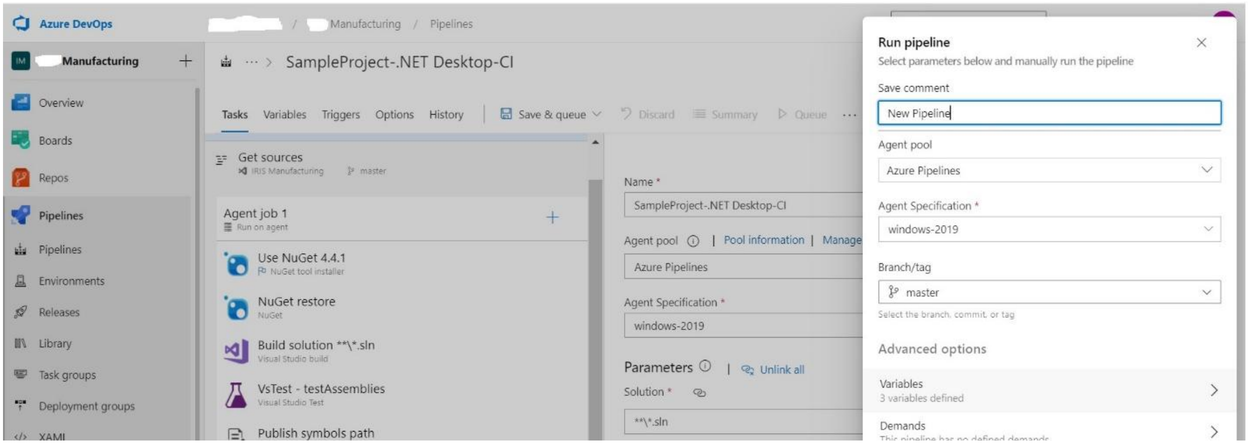
Name *
SampleProject-.NET Desktop-CI

Agent pool | Pool information | Manage
Azure Pipelines

Agent Specification *
windows-2019

Parameters | Unlink all
Solution *
***.sln

Step 8: A project is created with the name provided.



Run pipeline
Select parameters below and manually run the pipeline

Save comment
New Pipeline

Agent pool
Azure Pipelines

Agent Specification *
windows-2019

Branch/tag
master
Select the branch, commit, or tag

Advanced options

Variables
3 variables defined

Demands
This pipeline has no defined demands

SampleProject-.NET Desktop-CI

Get sources
IRIS Manufacturing master

Agent job 1
Run on agent

Use NuGet 4.4.1
NuGet tool installer

NuGet restore
NuGet

Build solution ***.sln
Visual Studio build

VsTest - testAssemblies
Visual Studio Test

Publish symbols path

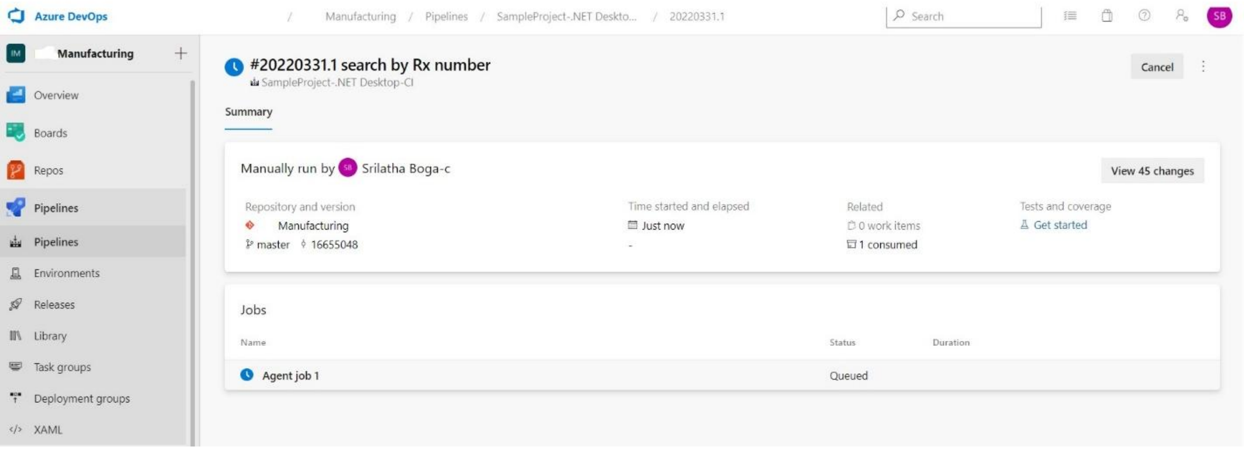
Name *
SampleProject-.NET Desktop-CI

Agent pool | Pool information | Manage
Azure Pipelines

Agent Specification *
windows-2019

Parameters | Unlink all
Solution *
***.sln

Step 9: By clicking on the "Agent Job" under Jobs section, we can review the builds dashboard and all the steps.



Azure DevOps / Manufacturing / Pipelines / SampleProject-.NET Deskto... / 20220331.1

Manufacturing

Overview
Boards
Repos
Pipelines
Pipelines
Environments
Releases
Library
Task groups
Deployment groups
XAML

#20220331.1 search by Rx number
SampleProject-.NET Desktop-CI

Summary

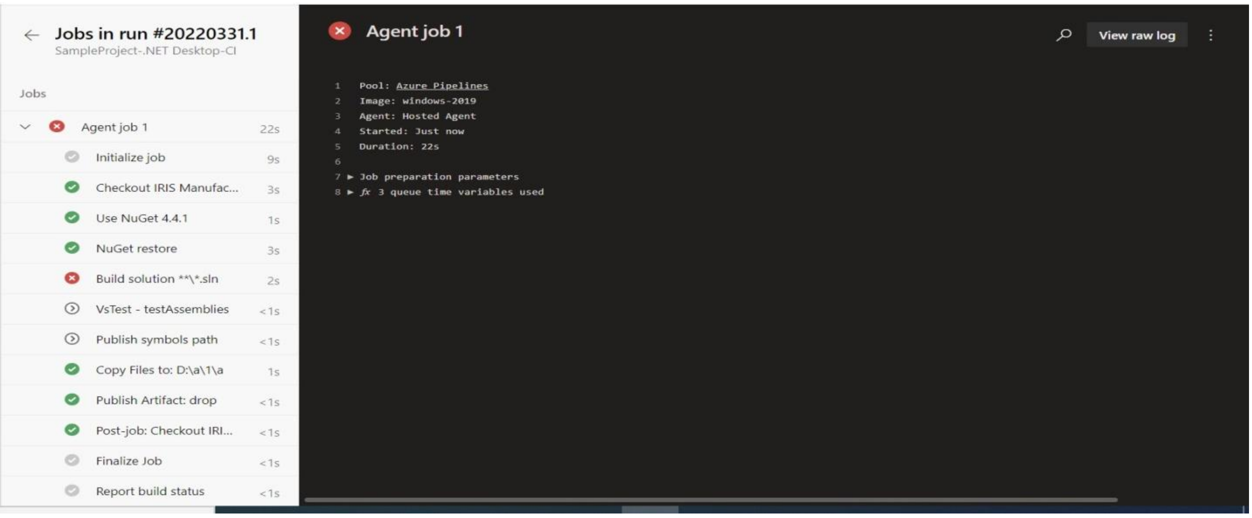
Manually run by Srilatha Boga-c

| Repository and version | Time started and elapsed | Related | Tests and coverage |
|---|---|---|---|
| Manufacturing | Just now | 0 work items | Get started |
| master 16655048 | - | 1 consumed | |

Jobs

| Name | Status | Duration |
|---|---|---|
| Agent job 1 | Queued | |

Step 10: If the build is success all the steps will be show with a green check else a red error message.



Jobs in run #20220331.1
SampleProject-.NET Desktop-CI

Jobs

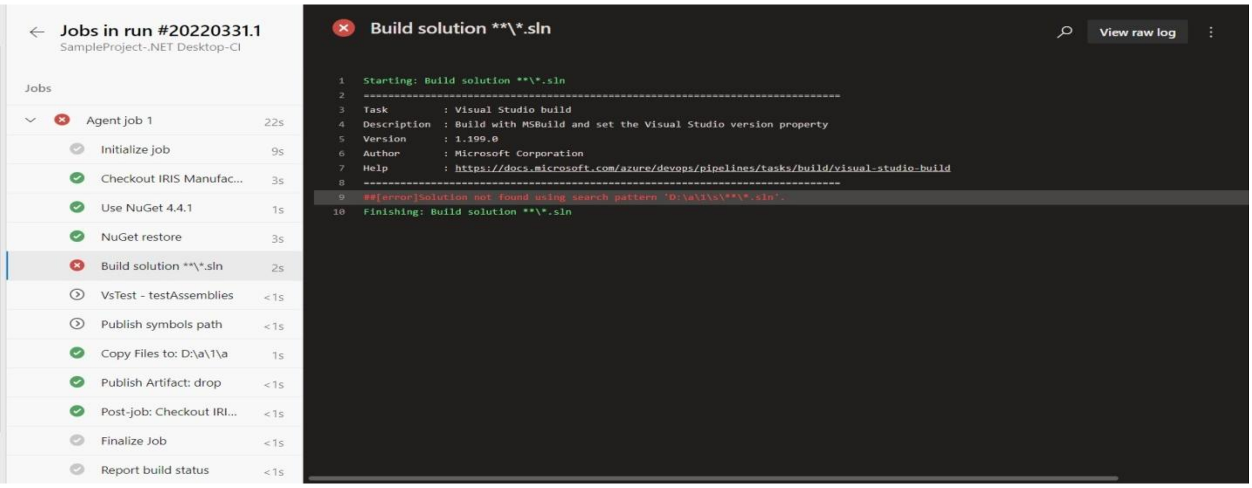
| | |
|---|---|
| Agent job 1 | 22s |
| Initialize job | 9s |
| Checkout IRIS Manufac... | 3s |
| Use NuGet 4.4.1 | 1s |
| NuGet restore | 3s |
| Build solution ***.sln | 2s |
| VsTest - testAssemblies | <1s |
| Publish symbols path | <1s |
| Copy Files to: D:\a\1\a | 1s |
| Publish Artifact: drop | <1s |
| Post-job: Checkout IRI... | <1s |
| Finalize Job | <1s |
| Report build status | <1s |

Agent job 1

```
Pool: Azure Pipelines
Image: windows-2019
Agent: Hosted Agent
Started: Just now
Duration: 22s

► Job preparation parameters
► fx 3 queue time variables used
```

Step 11: This project can be cloned from the "Repos" section!



Jobs in run #20220331.1
SampleProject-.NET Desktop-CI

Jobs

| | |
|---|---|
| Agent job 1 | 22s |
| Initialize job | 9s |
| Checkout IRIS Manufac... | 3s |
| Use NuGet 4.4.1 | 1s |
| NuGet restore | 3s |
| Build solution ***.sln | 2s |
| VsTest - testAssemblies | <1s |
| Publish symbols path | <1s |
| Copy Files to: D:\a\1\a | 1s |
| Publish Artifact: drop | <1s |
| Post-job: Checkout IRI... | <1s |
| Finalize Job | <1s |
| Report build status | <1s |

Build solution ***.sln

```
Starting: Build solution **\*.sln
==============================================================================
Task         : Visual Studio build
Description  : Build with MSBuild and set the Visual Studio version property
Version      : 1.199.0
Author       : Microsoft Corporation
Help         : https://docs.microsoft.com/azure/devops/pipelines/tasks/build/visual-studio-build
==============================================================================
##[error]Solution not found using search pattern 'D:\a\1\s\**\*.sln'.
Finishing: Build solution **\*.sln
```