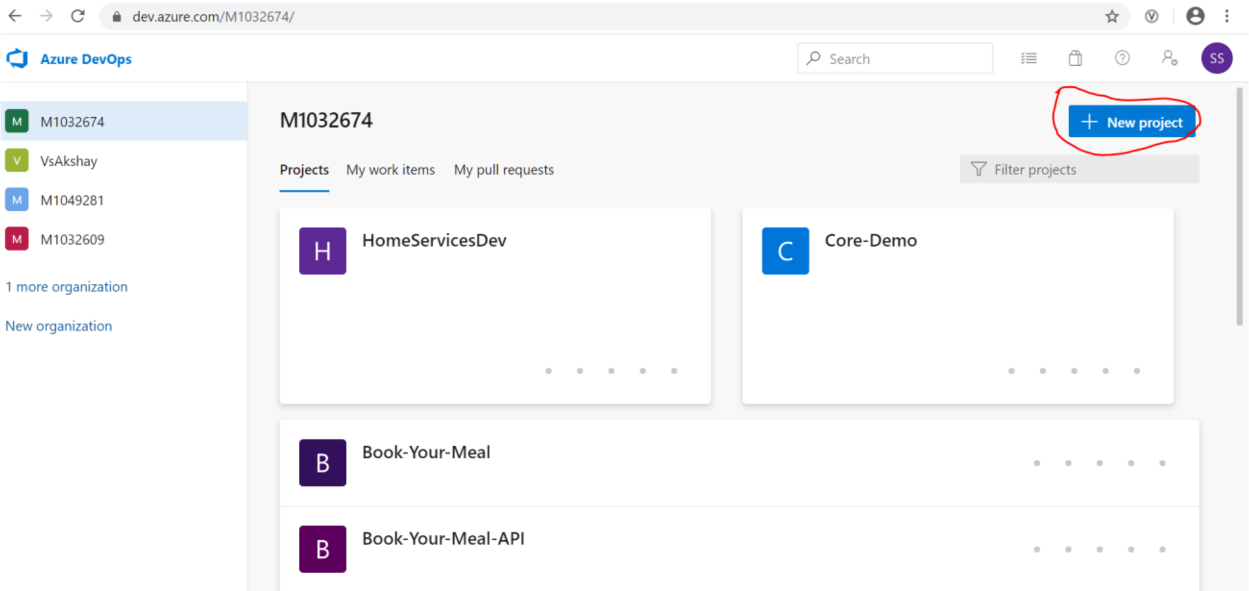
Shreeya Soni
M1032674

Dotnet Core CI-CD using Azure Repo

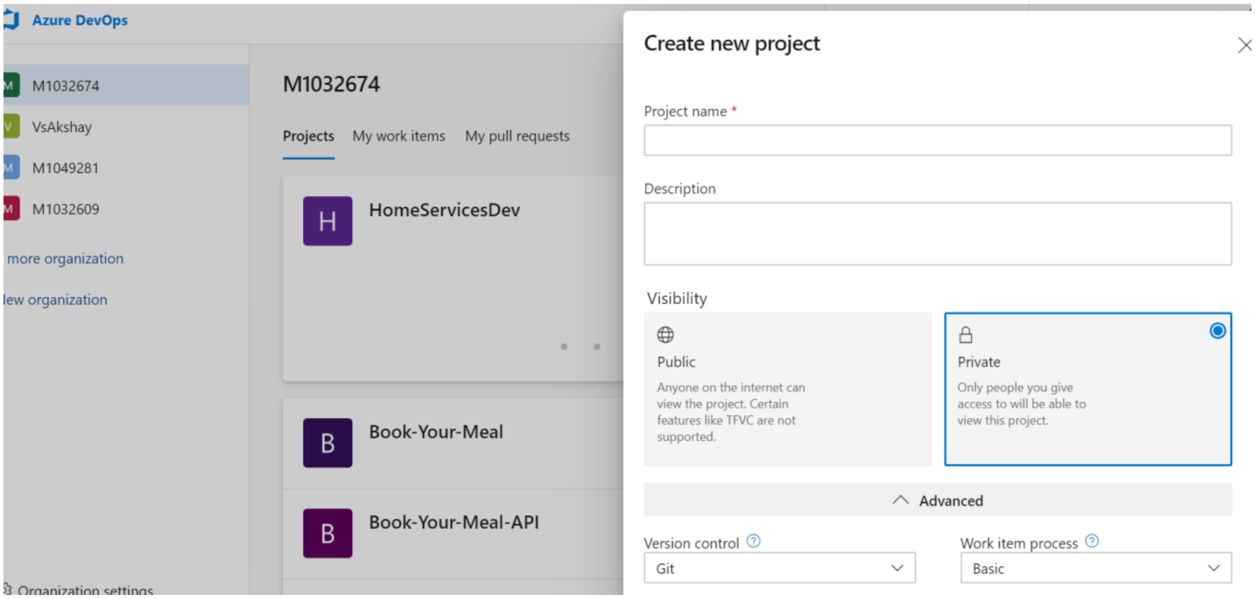
.net Core CI Steps

Go to https://dev.azure.com/

First create new project as shown below.



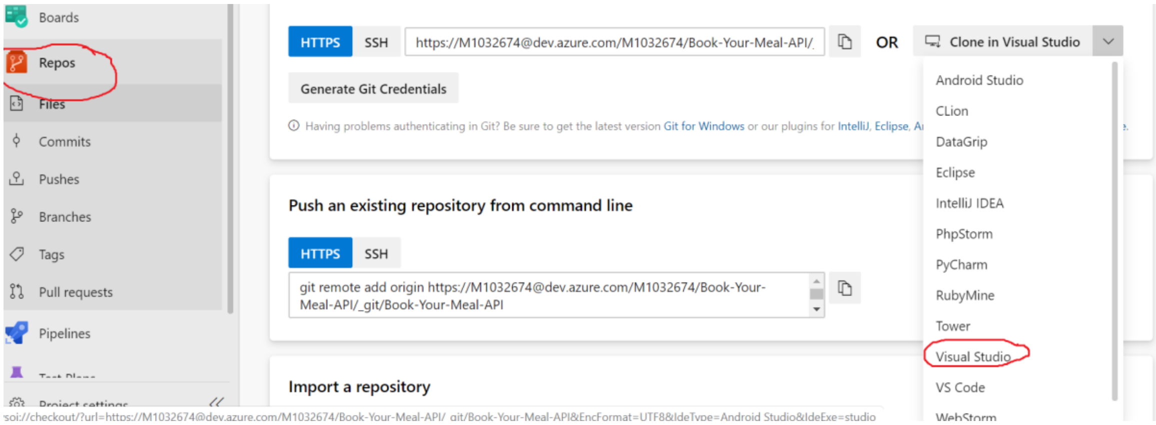
After you click on create project, you get below page, write project name, select private. In advance select Git and basic.



Version control: Git

Work item process: Basic

After successfully creation of project,Click on Repo from the left menu as shown below.Clone the Repository in VS by selecting clone in VS .So this is your repository where you will be pushing your code from VS.



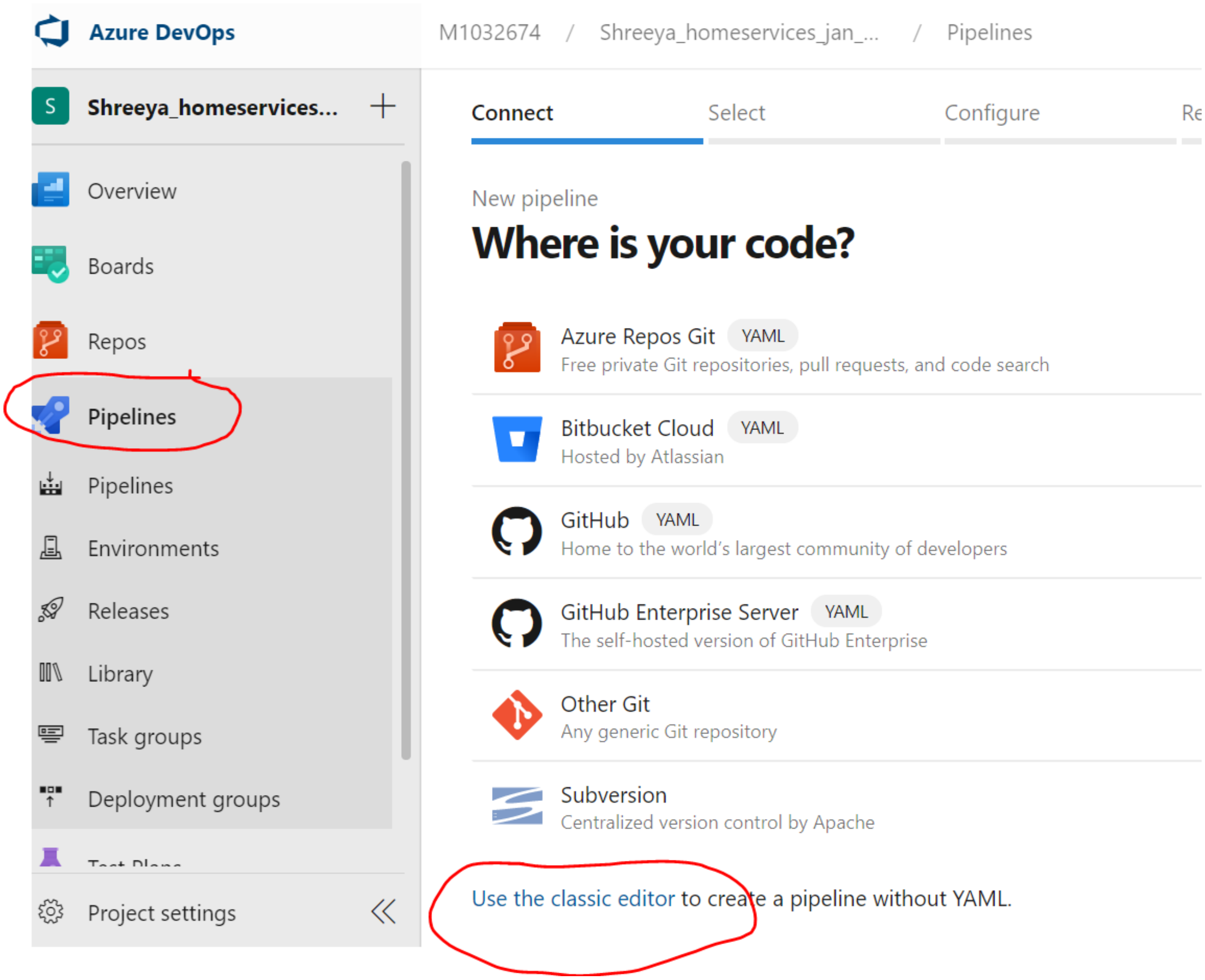
https://M1032674@dev.azure.com/M1032674/Book-Your-Meal-API/_

Having problems authenticating in Git? Be sure to get the latest version Git for Windows or our plugins for IntelliJ, Eclipse, A

```
git remote add origin https://M1032674@dev.azure.com/M1032674/Book-Your-Meal-API/_git/Book-Your-Meal-API
```

soi://checkout/?url=https://M1032674@dev.azure.com/M1032674/Book-Your-Meal-API/_git/Book-Your-Meal-API&EncFormat=UTF8&IdeType=Android Studio&IdeExe=studio

Now click on pipeline from left menu. You will the below page. Click on ’use classing editor’.



New pipeline

Where is your code?

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

Bitbucket Cloud YAML
Hosted by Atlassian

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

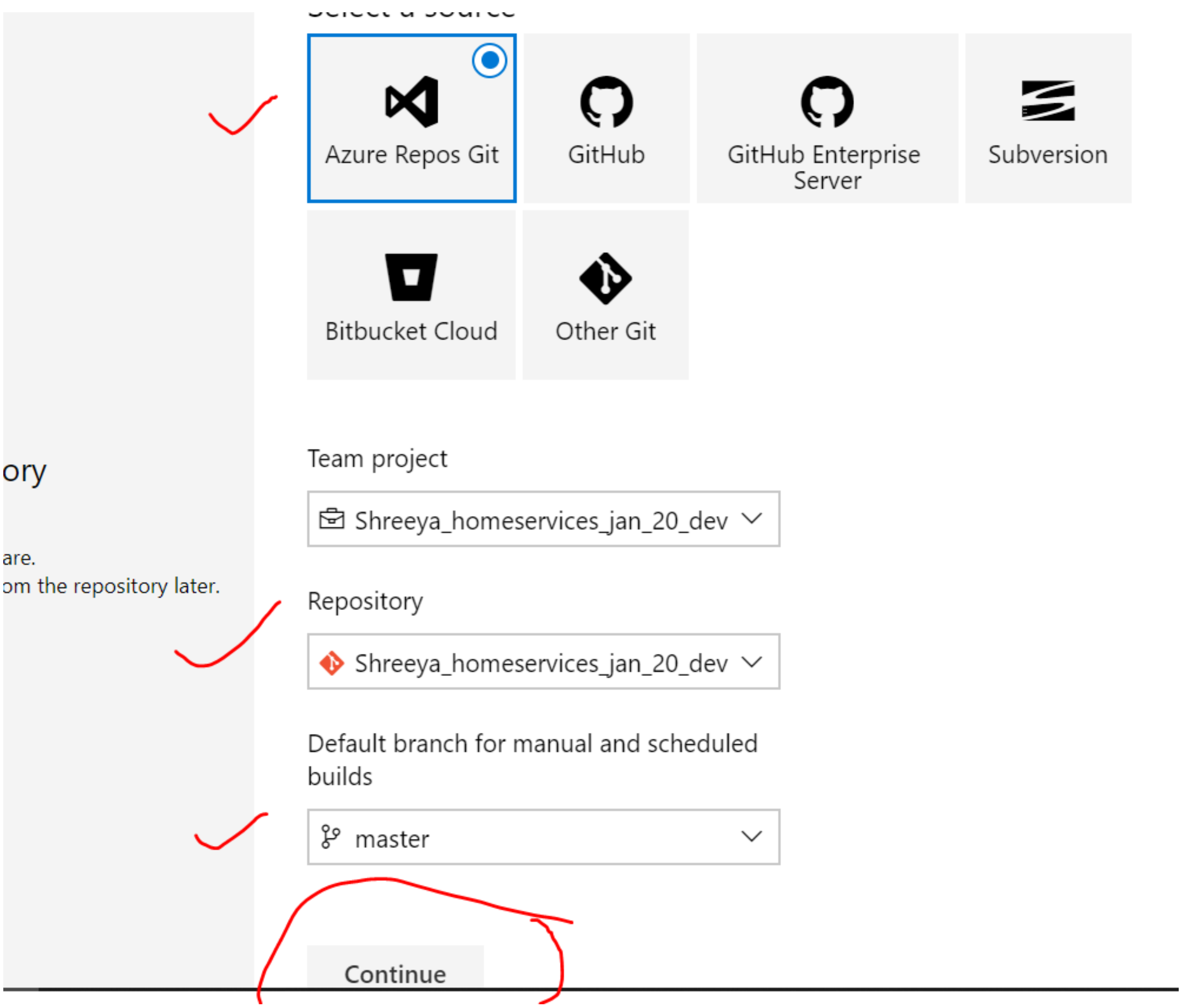
GitHub Enterprise Server YAML
The self-hosted version of GitHub Enterprise

Other Git
Any generic Git repository

Subversion
Centralized version control by Apache

Use the classic editor to create a pipeline without YAML.

You will get below page,Select Azure Repo Git as your repo is n Azure.Select the repo which you want to connect from this pipeline from the dropdown. [Note] It will the repo you have created inside the project.Click on continue.



Select a source

ory

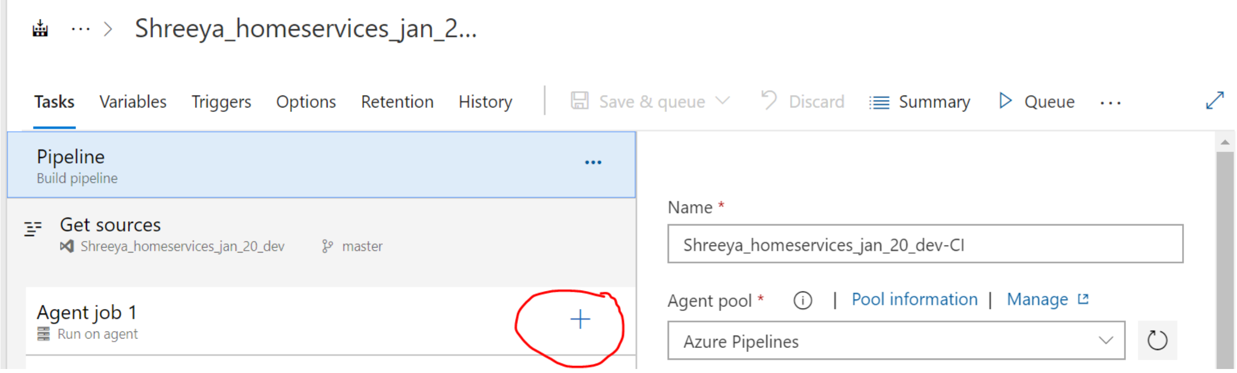
are.
om the repository later.

Team project: Shreeya_homeservices_jan_20_dev

Repository: Shreeya_homeservices_jan_20_dev

Default branch for manual and scheduled builds: master

After continue, you will get below page to create steps of Agent job. Click on plus sign as shown below.



Pipeline
Build pipeline

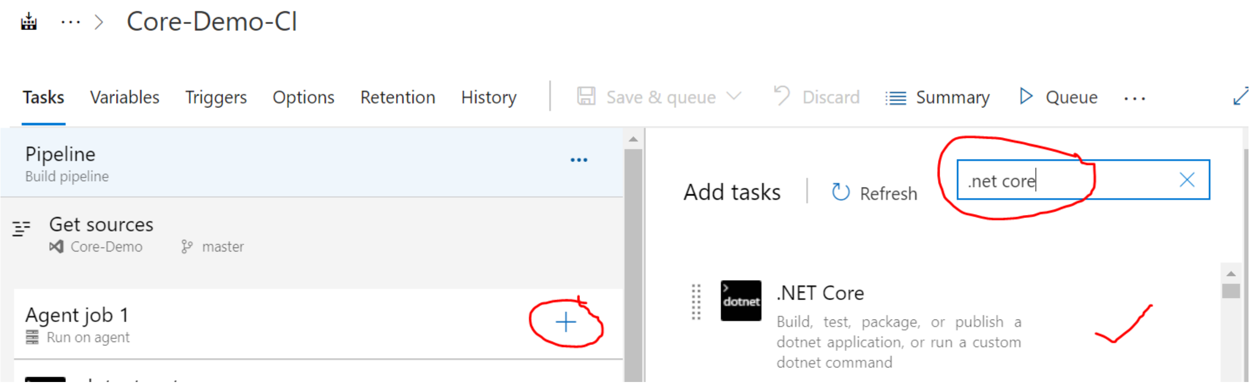
Get sources
Shreeya_homeservices_jan_20_dev master

Agent job 1
Run on agent

Name *: Shreeya_homeservices_jan_20_dev-CI

Agent pool * | Pool information | Manage: Azure Pipelines

Once you click on plus sign, you will get below pop-up to seach.Search .net core here ,select it and click on Add.[Note:Do not worry about the names,here project name is different as I changed it later]

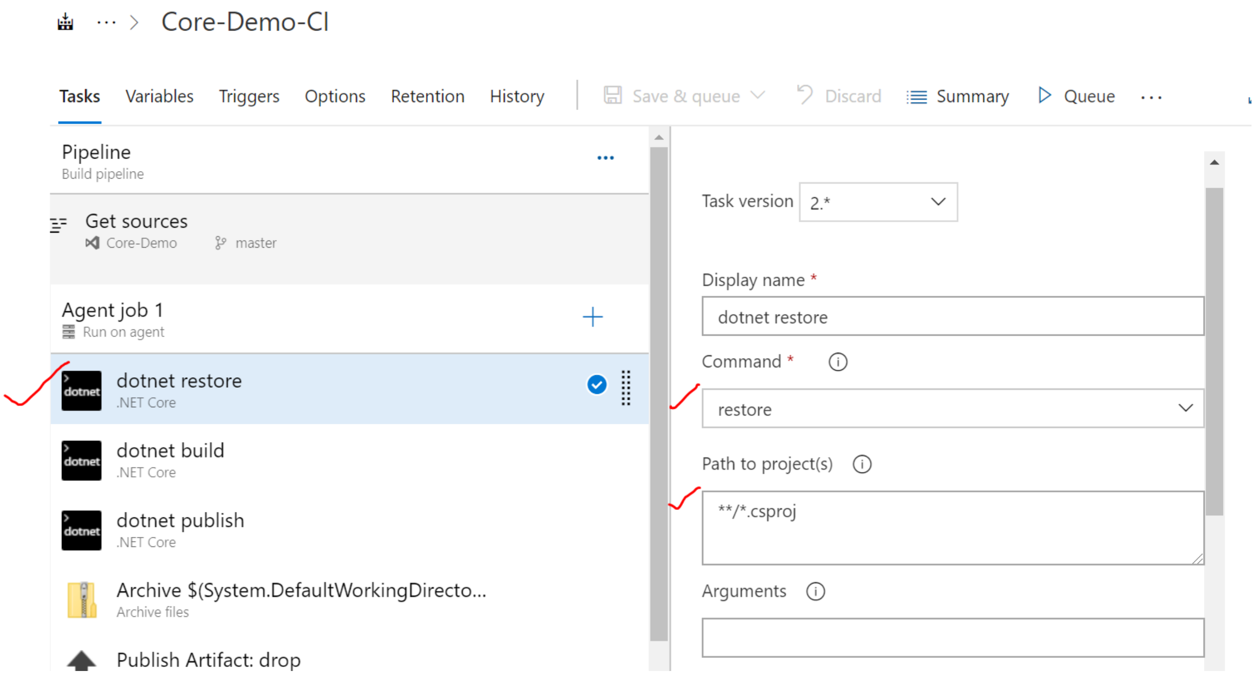


.net core

.NET Core

Build, test, package, or publish a dotnet application, or run a custom dotnet command

Select the added task and write the following commands as shown below.



Task version: 2.*

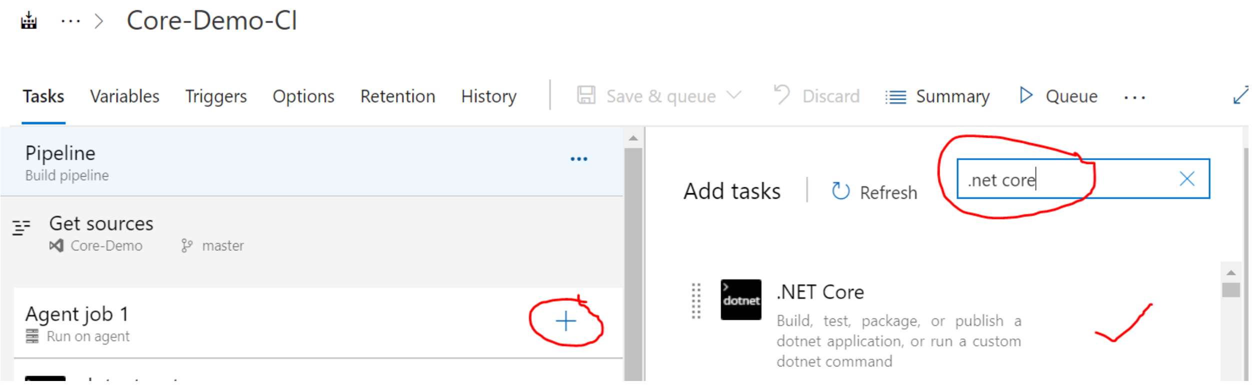
Display name *: dotnet restore

Command *: restore

Path to project(s): **/*.csproj

Arguments:

Once again you click on plus sign, you will get below pop-up to seach.Search .net core here ,select it and click on Add.



Core-Demo-CI

Tasks | Variables | Triggers | Options | Retention | History | Save & queue | Discard | Summary | Queue

Pipeline
Build pipeline

Get sources
Core-Demo master

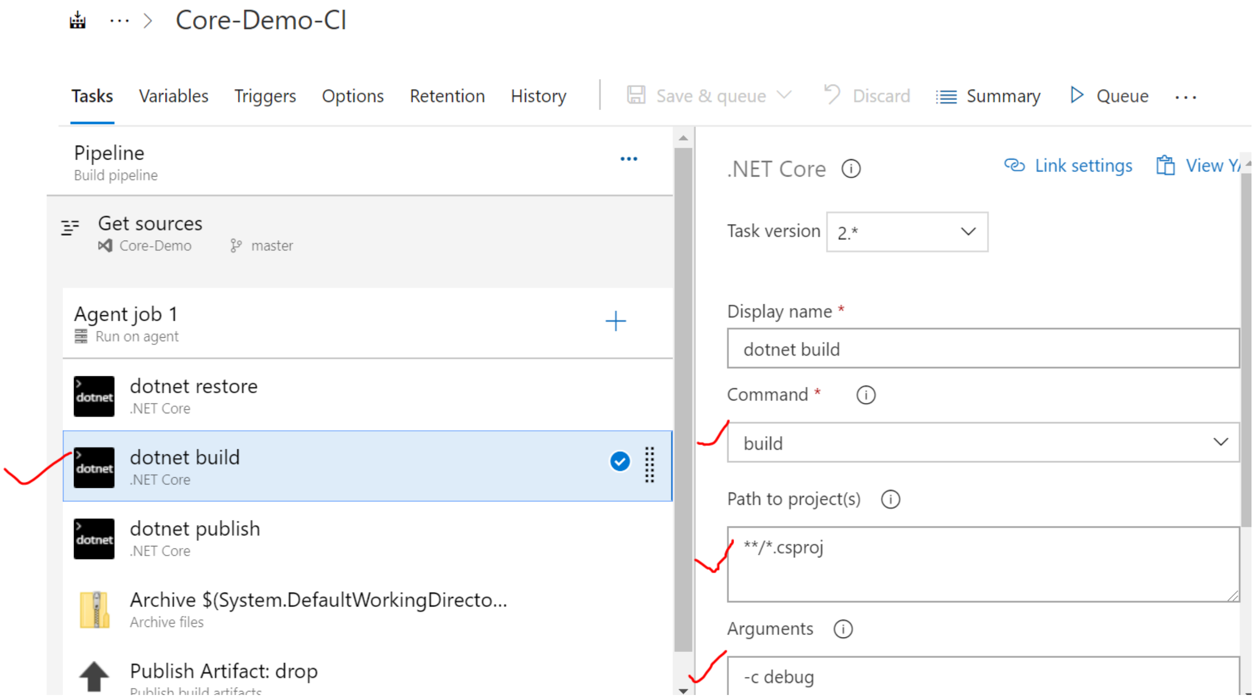
Agent job 1
Run on agent

Add tasks Refresh

Search: .net core

.NET Core
Build, test, package, or publish a dotnet application, or run a custom dotnet command

Select the new task added, and write the below commands as shown.



Core-Demo-CI

Tasks | Variables | Triggers | Options | Retention | History | Save & queue | Discard | Summary | Queue

Pipeline
Build pipeline

Get sources
Core-Demo master

Agent job 1
Run on agent

- dotnet restore (.NET Core)
- dotnet build (.NET Core)
- dotnet publish (.NET Core)
- Archive $(System.DefaultWorkingDirecto... (Archive files)
- Publish Artifact: drop (Publish build artifacts)

.NET Core

Link settings | View Y

Task version: 2.*

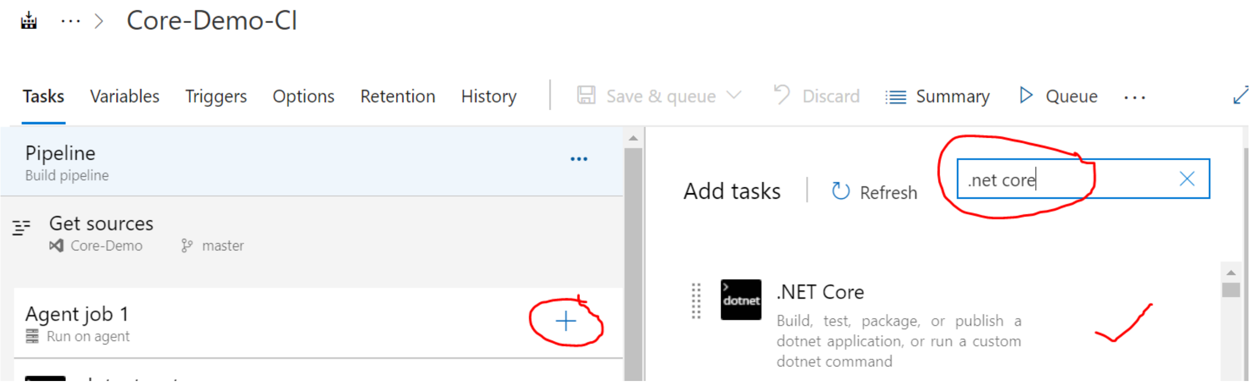
Display name *: dotnet build

Command *: build

Path to project(s): **/*.csproj

Arguments: -c debug

Once again you click on plus sign, you will get below pop-up to seach.Search .net core here ,select it and click on Add.

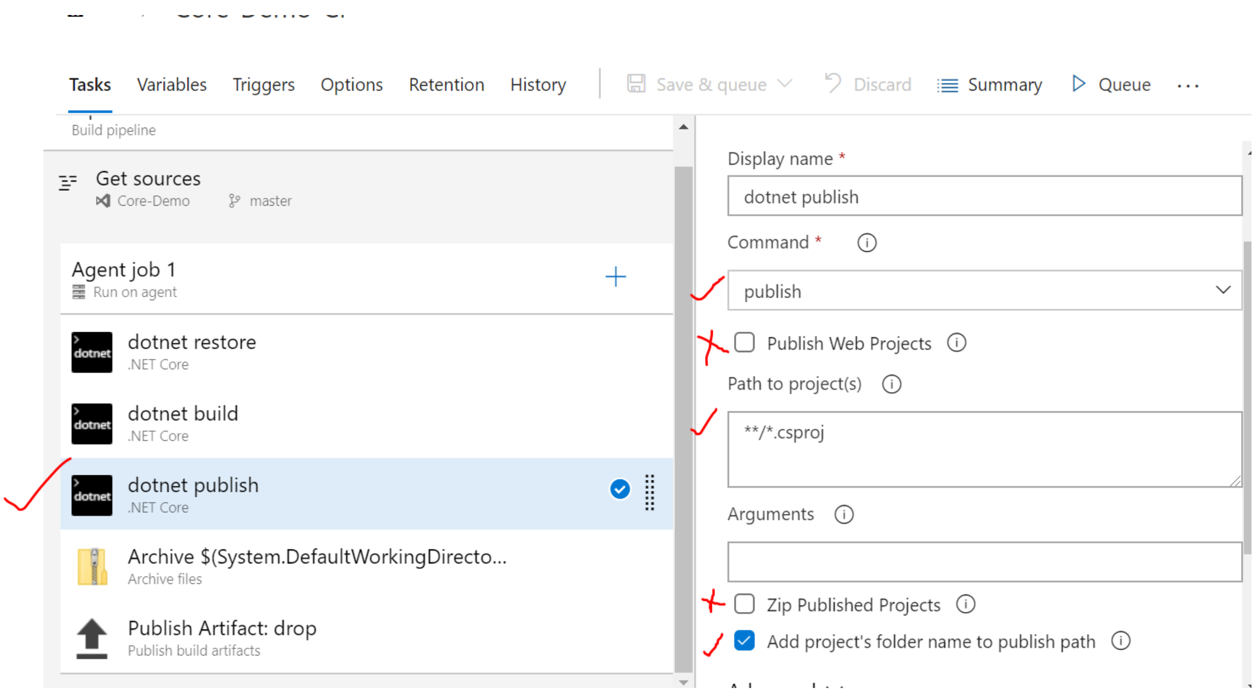


Core-Demo-CI

Search: .net core

.NET Core — Build, test, package, or publish a dotnet application, or run a custom dotnet command

Select the newly added task and write the below command as shown.



Display name *: dotnet publish

Command *: publish

□ Publish Web Projects

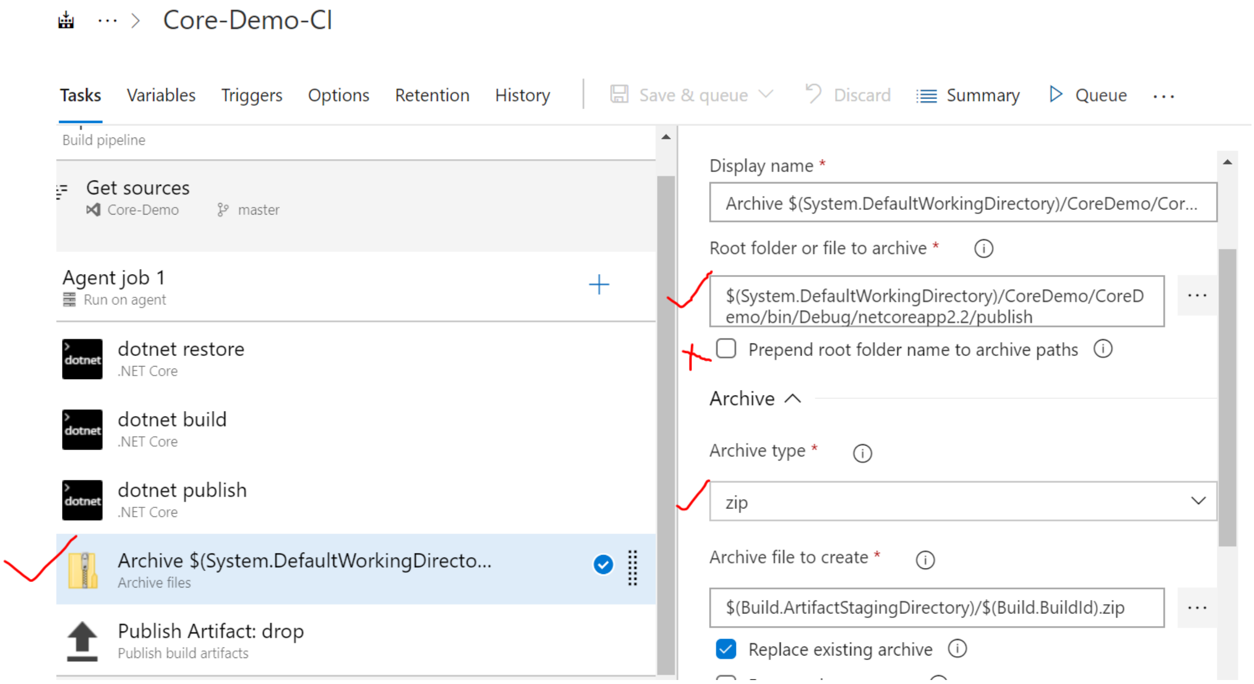
Path to project(s): **/*.csproj

Arguments:

□ Zip Published Projects

☑ Add project's folder name to publish path

Similary ,add one more task.Search 'Archive files' ,select it and add it.After adding add the below commands as shown.Build files will be generated in project folder/bin/Debug/netcoreapp2.2/publish,so it will be root folder here which we need to archive.



Get sources
Core-Demo master

Agent job 1
Run on agent

dotnet restore
.NET Core

dotnet build
.NET Core

dotnet publish
.NET Core

Archive $(System.DefaultWorkingDirecto...
Archive files

Publish Artifact: drop
Publish build artifacts

Display name *
Archive $(System.DefaultWorkingDirectory)/CoreDemo/Cor...

Root folder or file to archive *
$(System.DefaultWorkingDirectory)/CoreDemo/CoreDemo/bin/Debug/netcoreapp2.2/publish

☐ Prepend root folder name to archive paths

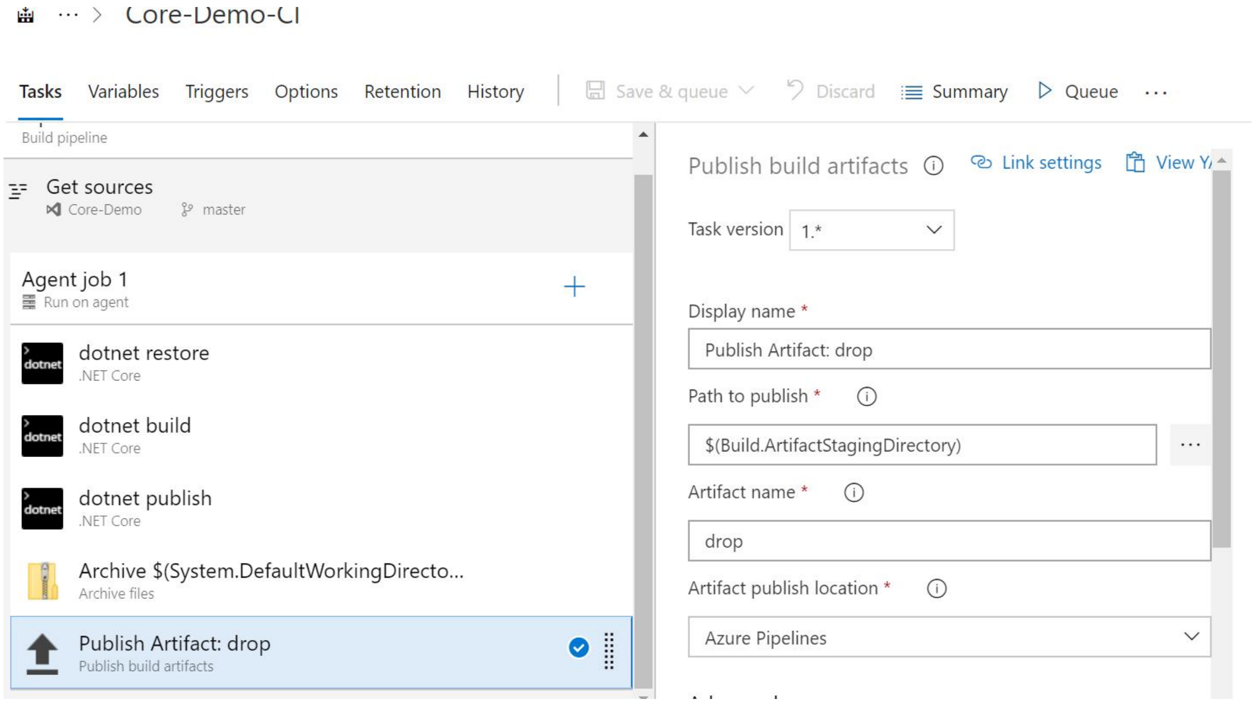
Archive

Archive type *
zip

Archive file to create *
$(Build.ArtifactStagingDirectory)/$(Build.BuildId).zip

☑ Replace existing archive

Similarly add one more task .Search ‘Publish build artifact’, select it and add it. Do not need to change anything here, let the default command as it is.



Get sources
Core-Demo master

Agent job 1
Run on agent

dotnet restore
.NET Core

dotnet build
.NET Core

dotnet publish
.NET Core

Archive $(System.DefaultWorkingDirecto...
Archive files

Publish Artifact: drop
Publish build artifacts

Publish build artifacts

Task version 1.*

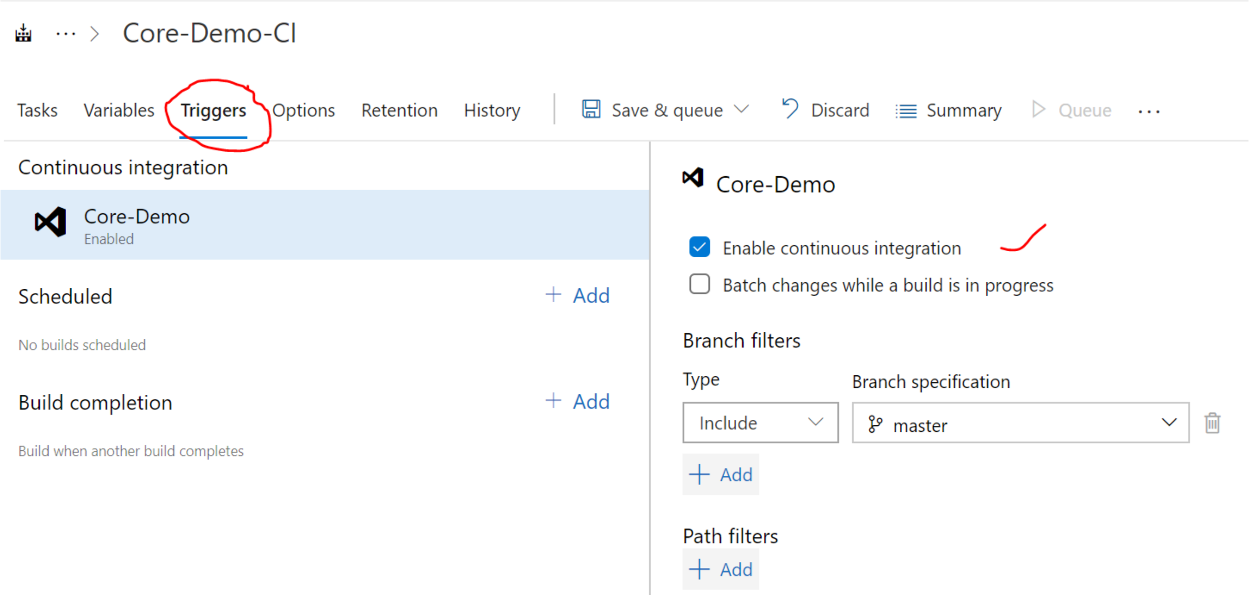
Display name *
Publish Artifact: drop

Path to publish *
$(Build.ArtifactStagingDirectory)

Artifact name *
drop

Artifact publish location *
Azure Pipelines

Now click on Triggers and enable continuous integration as shown below.



Continuous integration

Core-Demo
Enabled

Scheduled + Add

No builds scheduled

Build completion + Add

Build when another build completes

Core-Demo

☑ Enable continuous integration

☐ Batch changes while a build is in progress

Branch filters

Type	Branch specification
Include	master

+ Add

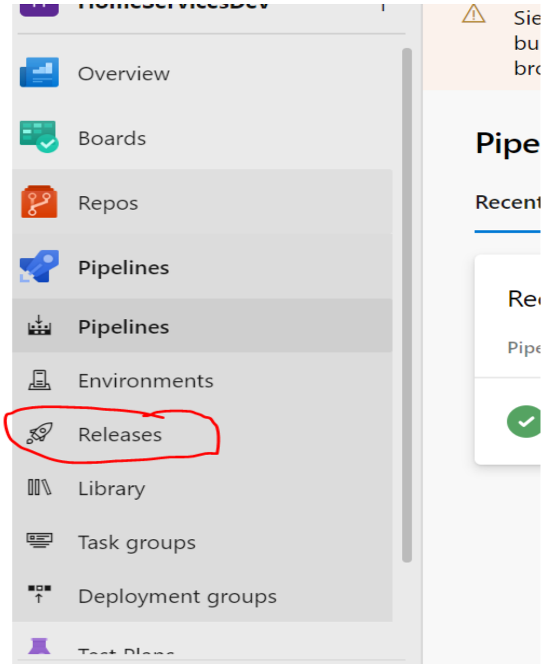
Path filters

+ Add

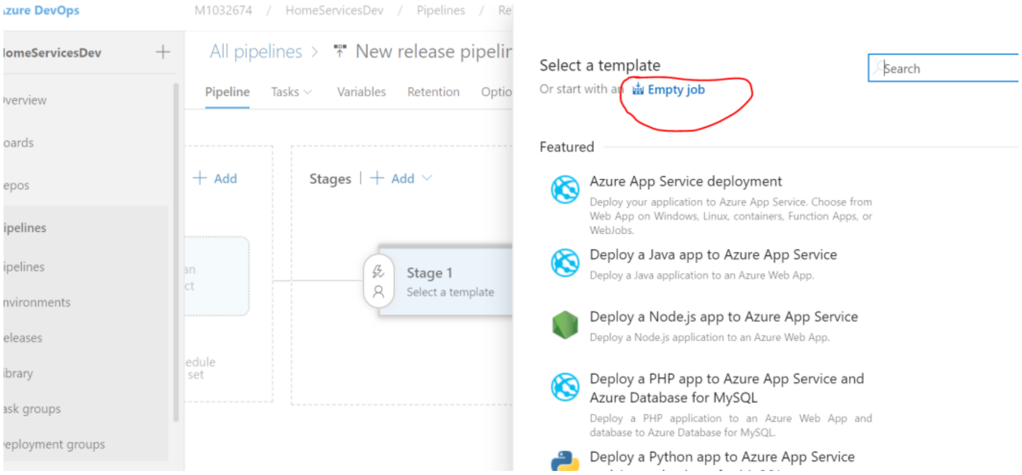
Shreeya Soni
M1032674

.net Core CD Steps

Now click on Releases from the left menu as shown below.



Once you selected release, then click on create new pipeline. You will get below page. Click on empty job.



Select a template

Or start with an Empty job

Featured

Azure App Service deployment
Deploy your application to Azure App Service. Choose from Web App on Windows, Linux, containers, Function Apps, or WebJobs.

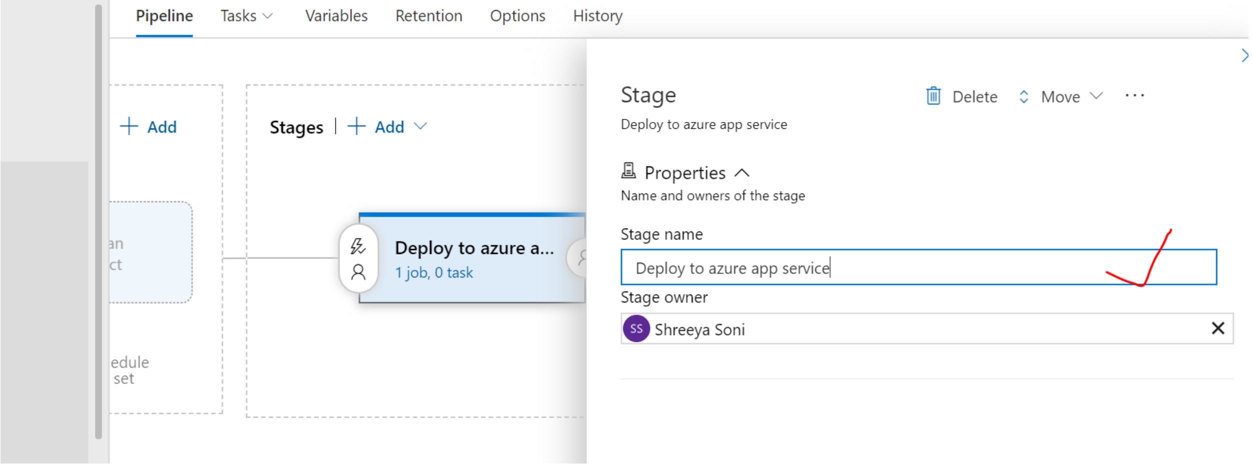
Deploy a Java app to Azure App Service
Deploy a Java application to an Azure Web App.

Deploy a Node.js app to Azure App Service
Deploy a Node.js application to an Azure Web App.

Deploy a PHP app to Azure App Service and Azure Database for MySQL
Deploy a PHP application to an Azure Web App and database to Azure Database for MySQL.

Deploy a Python app to Azure App Service

After it ,you will get below page ,change the stage name to Deploy to Azure app service and save it.

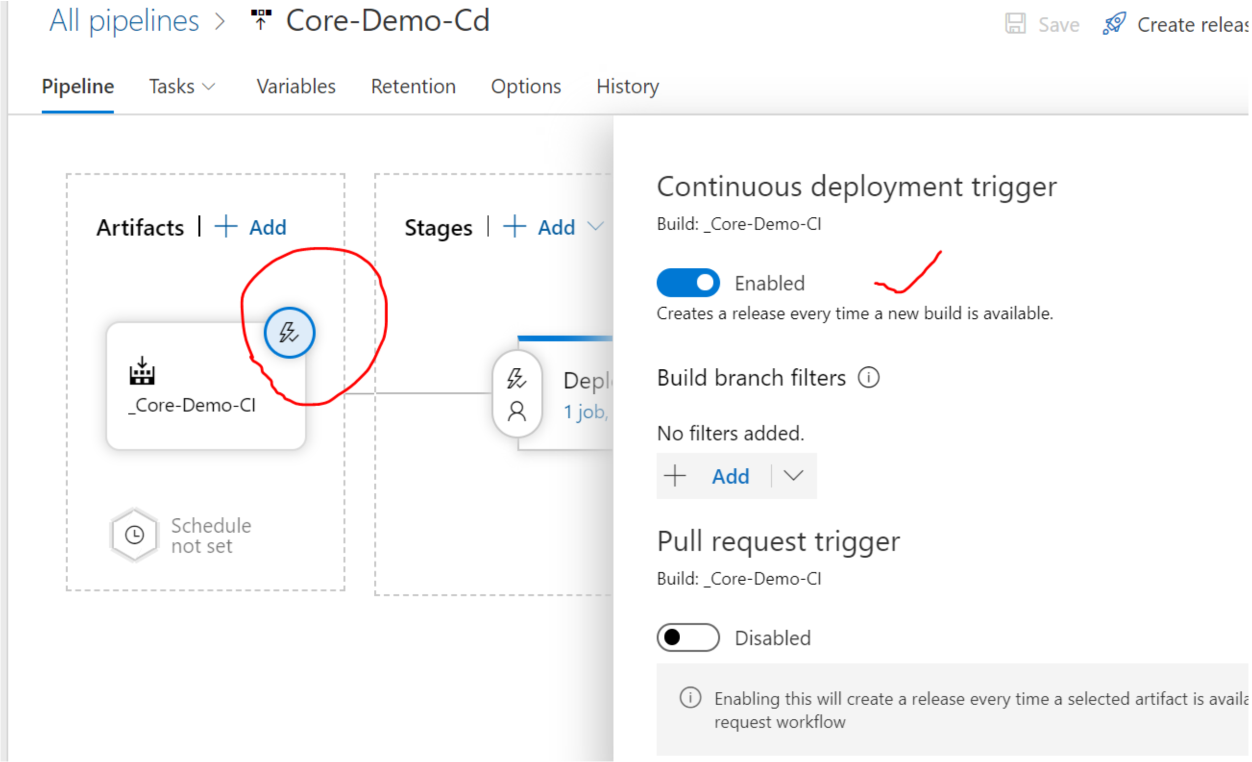


Stage
Deploy to azure app service

Stage name: Deploy to azure app service

Stage owner: Shreeya Soni

Now you will get below page, click on trigger as shown below and enable CD.



Continuous deployment trigger
Build: _Core-Demo-CI

Enabled
Creates a release every time a new build is available.

Build branch filters

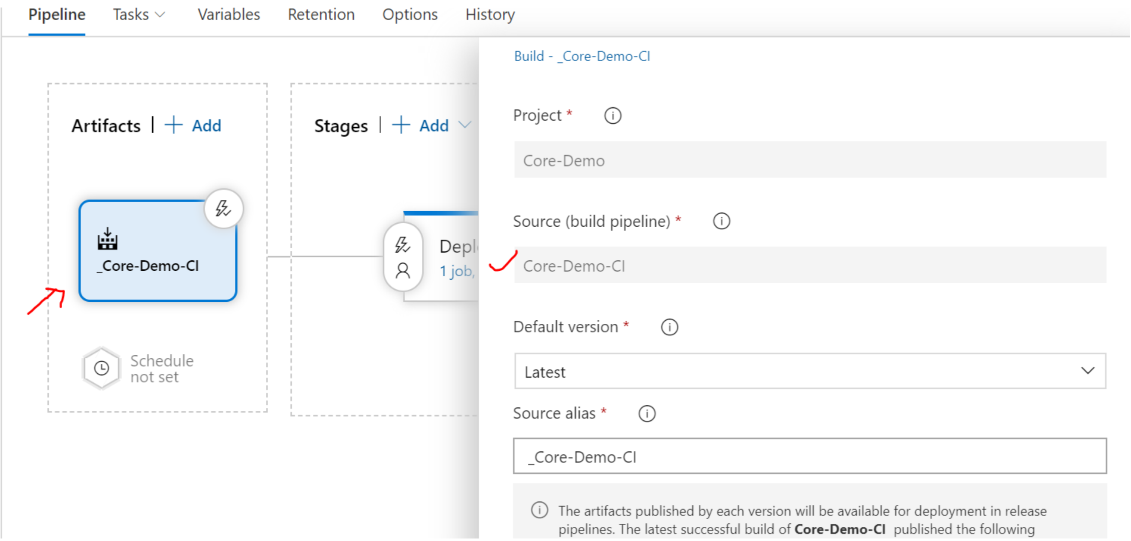
No filters added.

Pull request trigger
Build: _Core-Demo-CI

Disabled

Enabling this will create a release every time a selected artifact is availa
request workflow

Now click on artifacts,and select the source which will be your CI name.



Build - _Core-Demo-CI

Project * Core-Demo

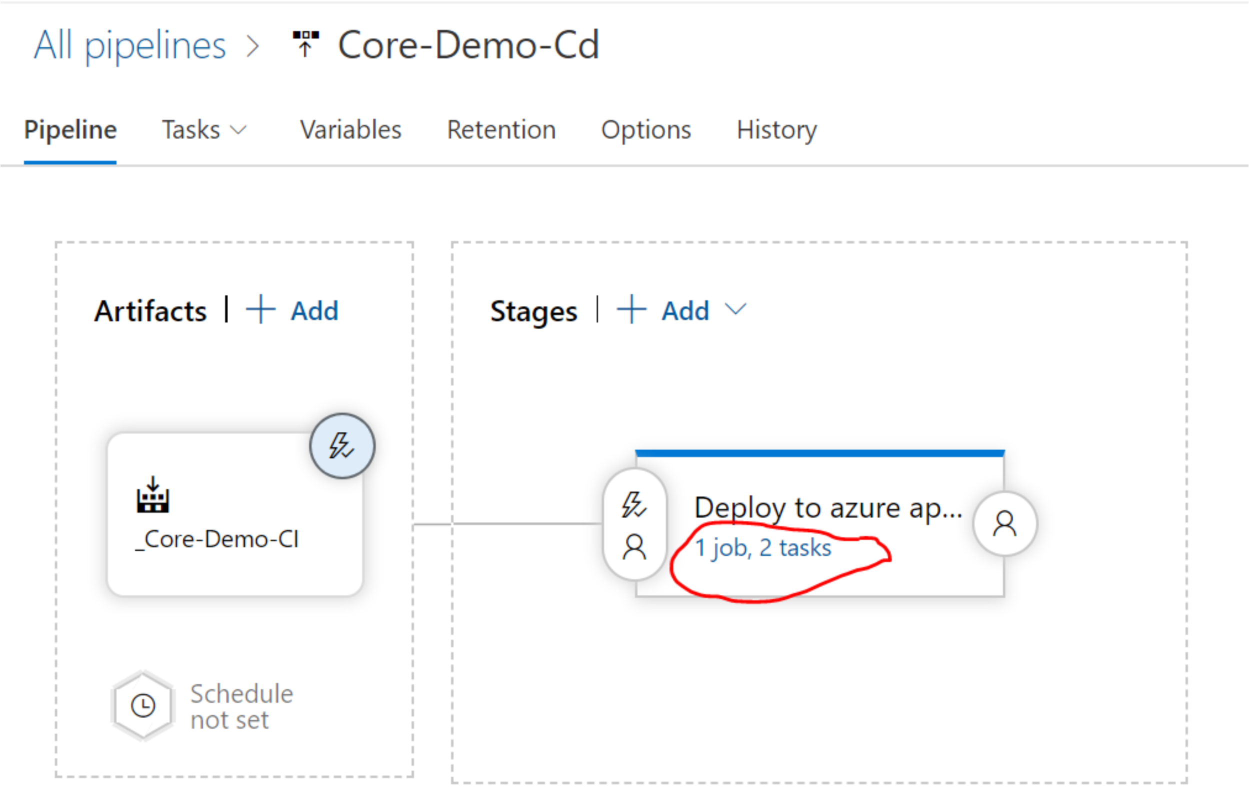
Source (build pipeline) * Core-Demo-CI

Default version * Latest

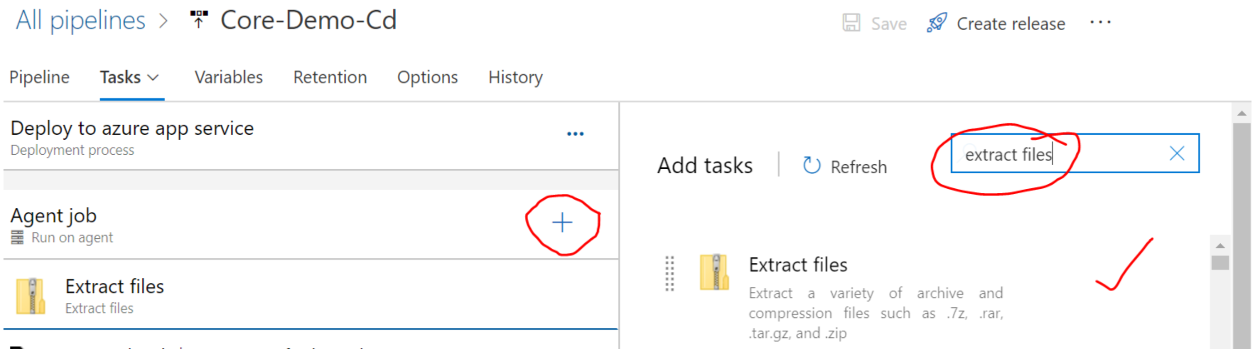
Source alias * _Core-Demo-CI

The artifacts published by each version will be available for deployment in release pipelines. The latest successful build of **Core-Demo-CI** published the following

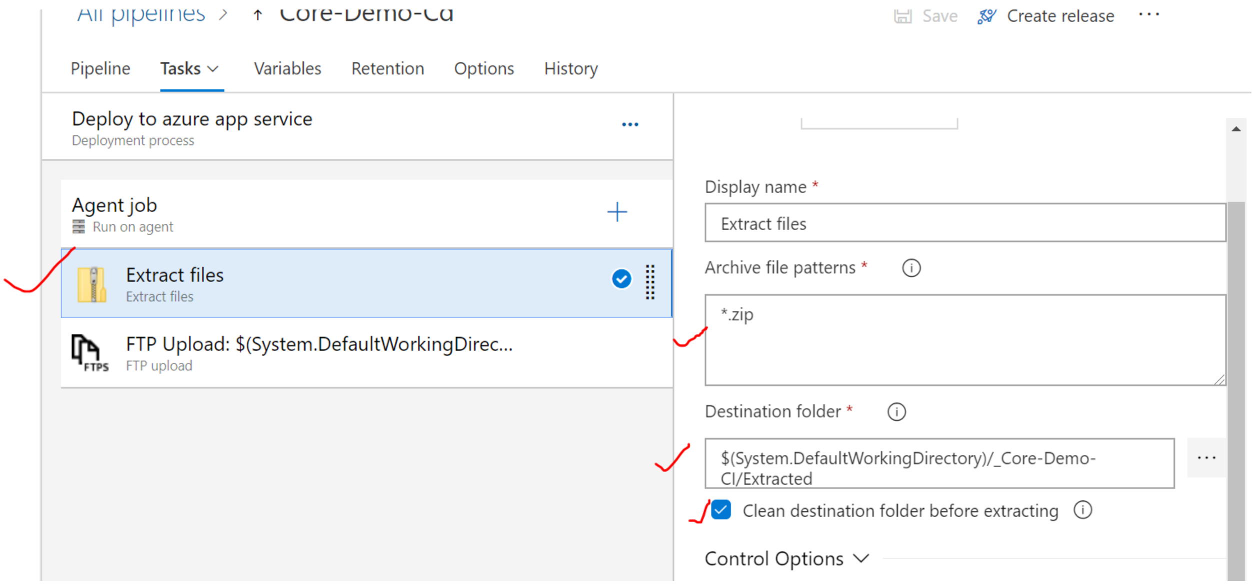
Now click on 1 job,2 tasks link as show below. For you it will show 0 tasks as there will not be any tasks



Once again you click on plus sign, you will get below pop-up to seach.Search for extract files,select and add it.



Now select extract files from the tasks and write the following commands as shown below. Go to three dots select the CI alias name given to ci pipeline. Then add the folder name extracted where you want to extract the files



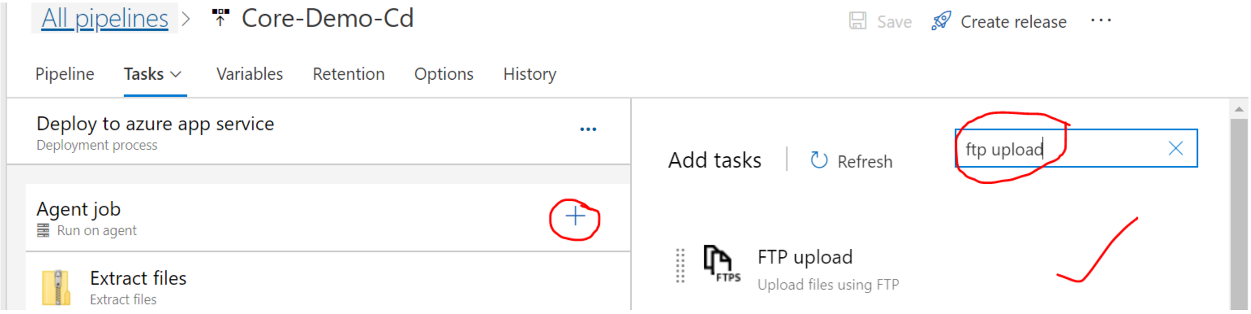
Display name *
Extract files

Archive file patterns *
*.zip

Destination folder *
$(System.DefaultWorkingDirectory)/_Core-Demo-CI/Extracted

☑ Clean destination folder before extracting

Once again you click on plus sign, you will get below pop-up to seach.Search for FTP Upload, select and add it to your task.



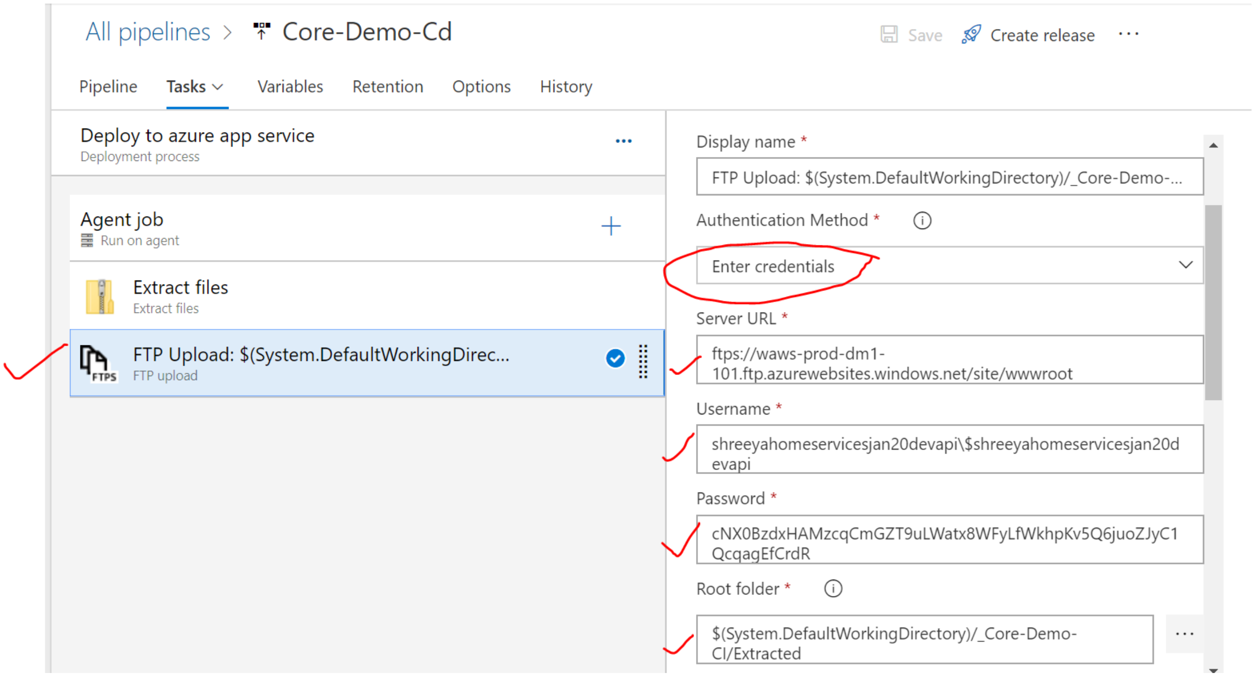
ftp upload

FTP upload
Upload files using FTP

Select FTP upload and write the below command as shown.

Here your root folder and will be same as destination folder given in extract files task above.

[Note]For FTP you first need to create app service in azure portal ,get the url, username and password from app service→deployment center→FTP.



Display name *
FTP Upload: $(System.DefaultWorkingDirectory)/_Core-Demo-...

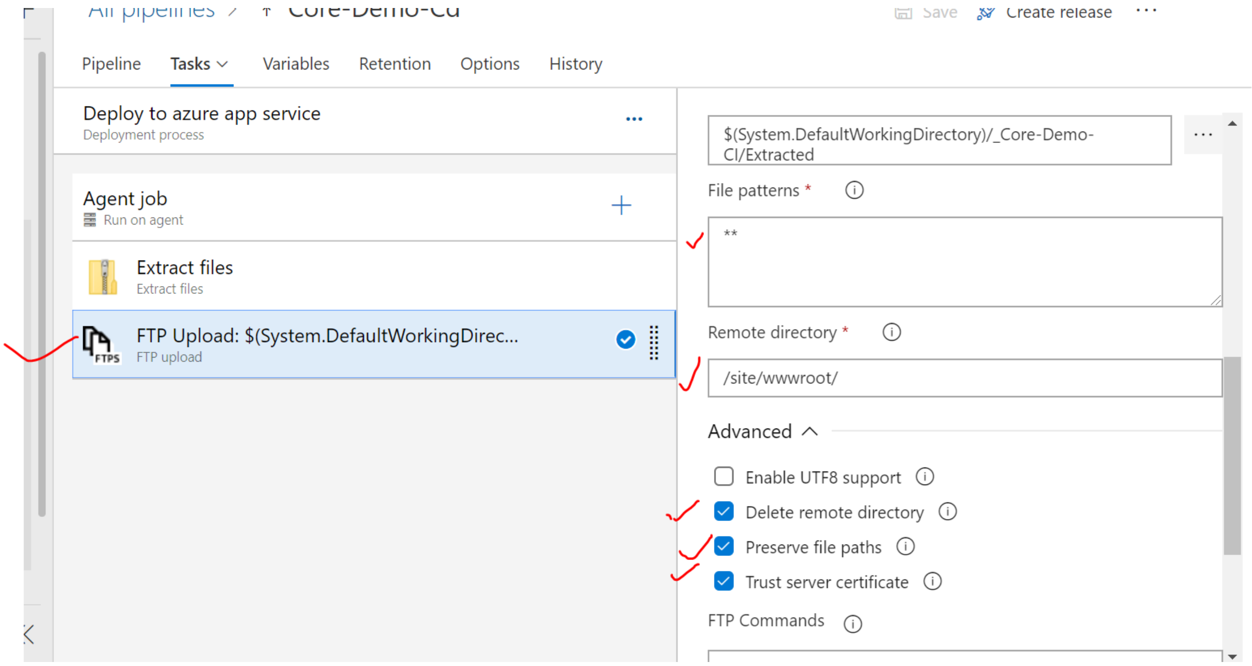
Authentication Method *
Enter credentials

Server URL *
ftps://waws-prod-dm1-101.ftp.azurewebsites.windows.net/site/wwwroot

Username *
shreeyahomeservicesjan20devapi\$shreeyahomeservicesjan20devapi

Password *
cNX0BzdxHAMzcqCmGZT9uLWatx8WFyLfWkhpKv5Q6juoZJyC1QcqagEfCrdR

Root folder *
$(System.DefaultWorkingDirectory)/_Core-Demo-CI/Extracted



Deploy to azure app service
Deployment process

Agent job
Run on agent

Extract files
Extract files

FTP Upload: $(System.DefaultWorkingDirec...
FTP upload

$(System.DefaultWorkingDirectory)/_Core-Demo-CI/Extracted

File patterns *

```
**
```

Remote directory *

/site/wwwroot/

Advanced

☐ Enable UTF8 support

☑ Delete remote directory

☑ Preserve file paths

☑ Trust server certificate

FTP Commands