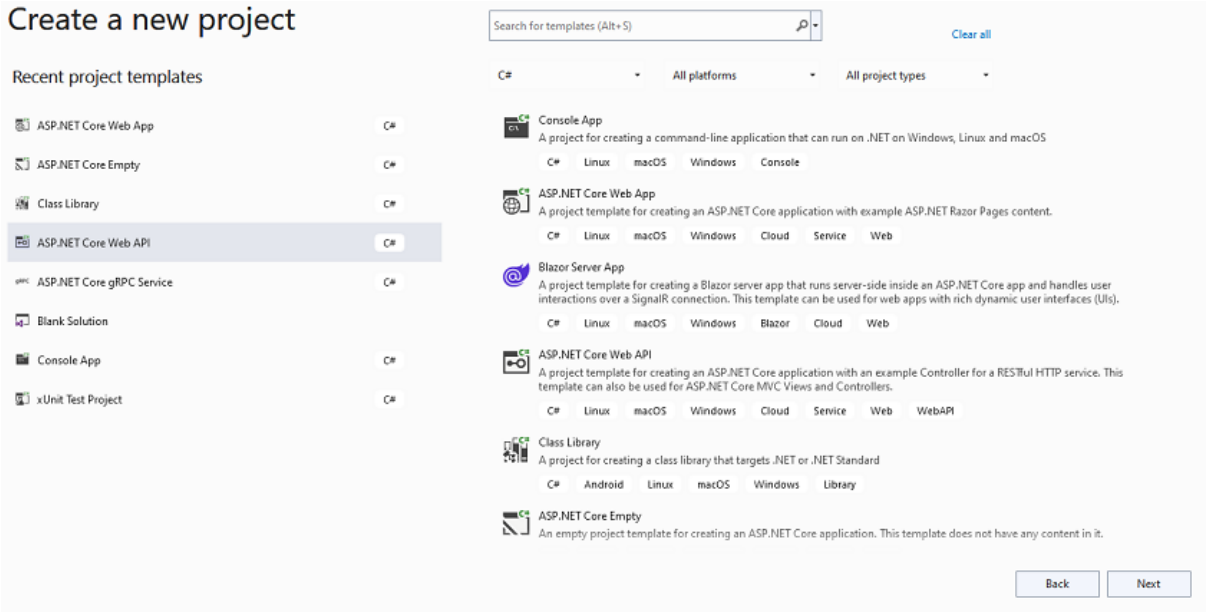
Step by Step Implementation of.NET Core Application

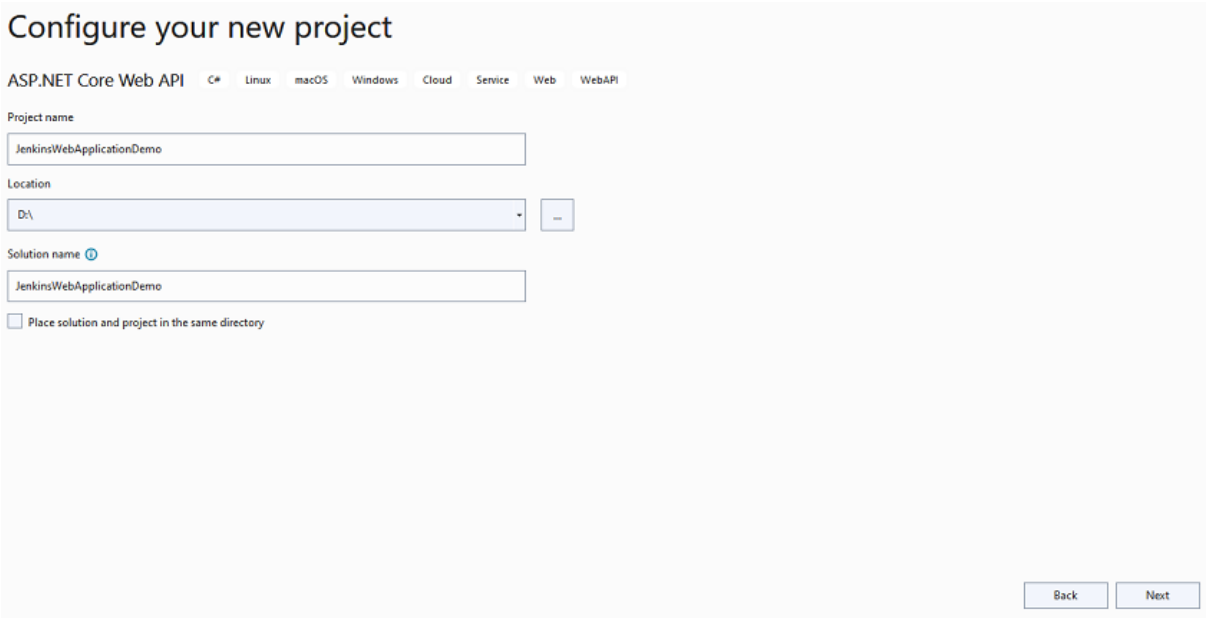
Step 1

Create a new Web API



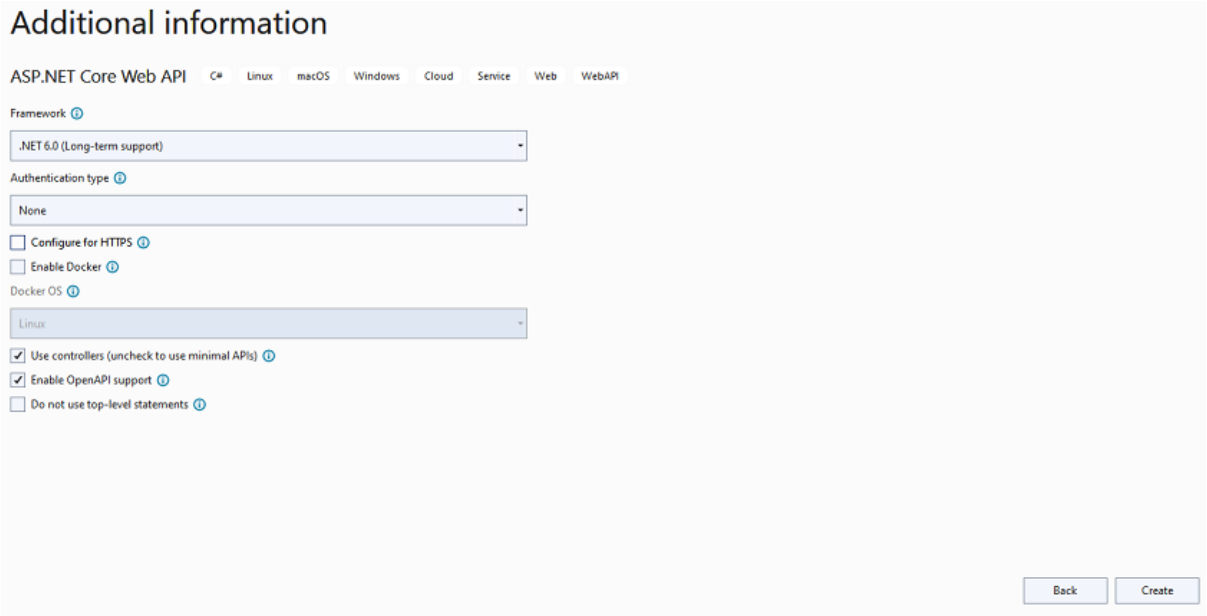
Step 2

Configure the application



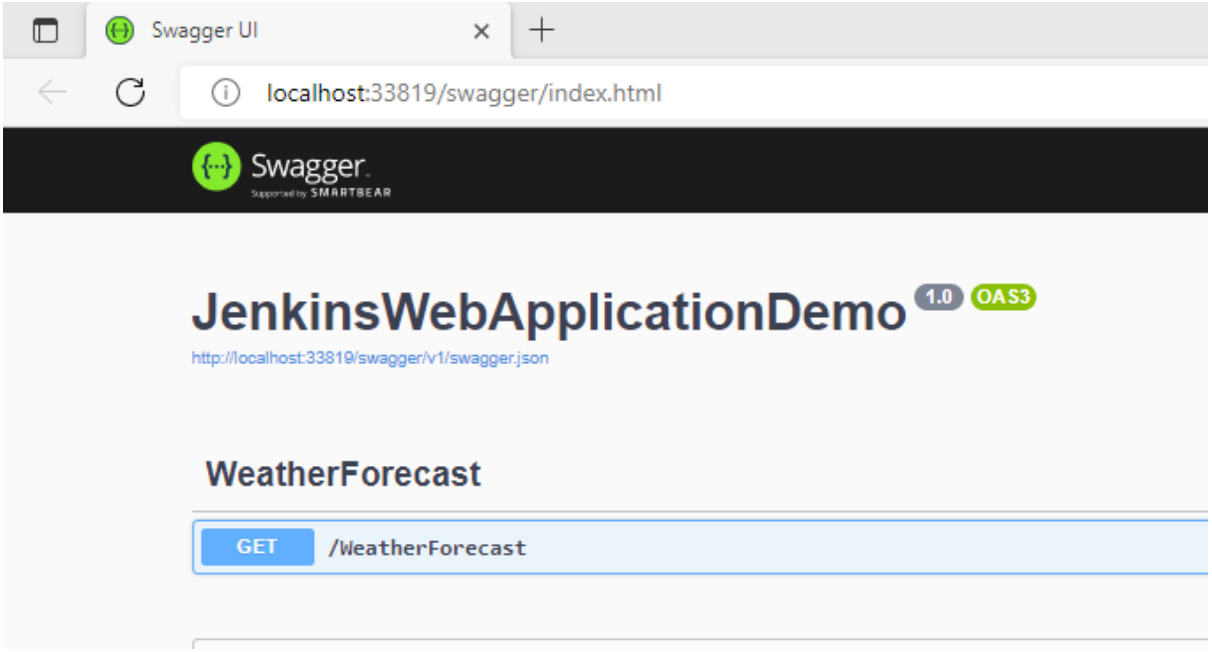
Step 3

Provide additional information



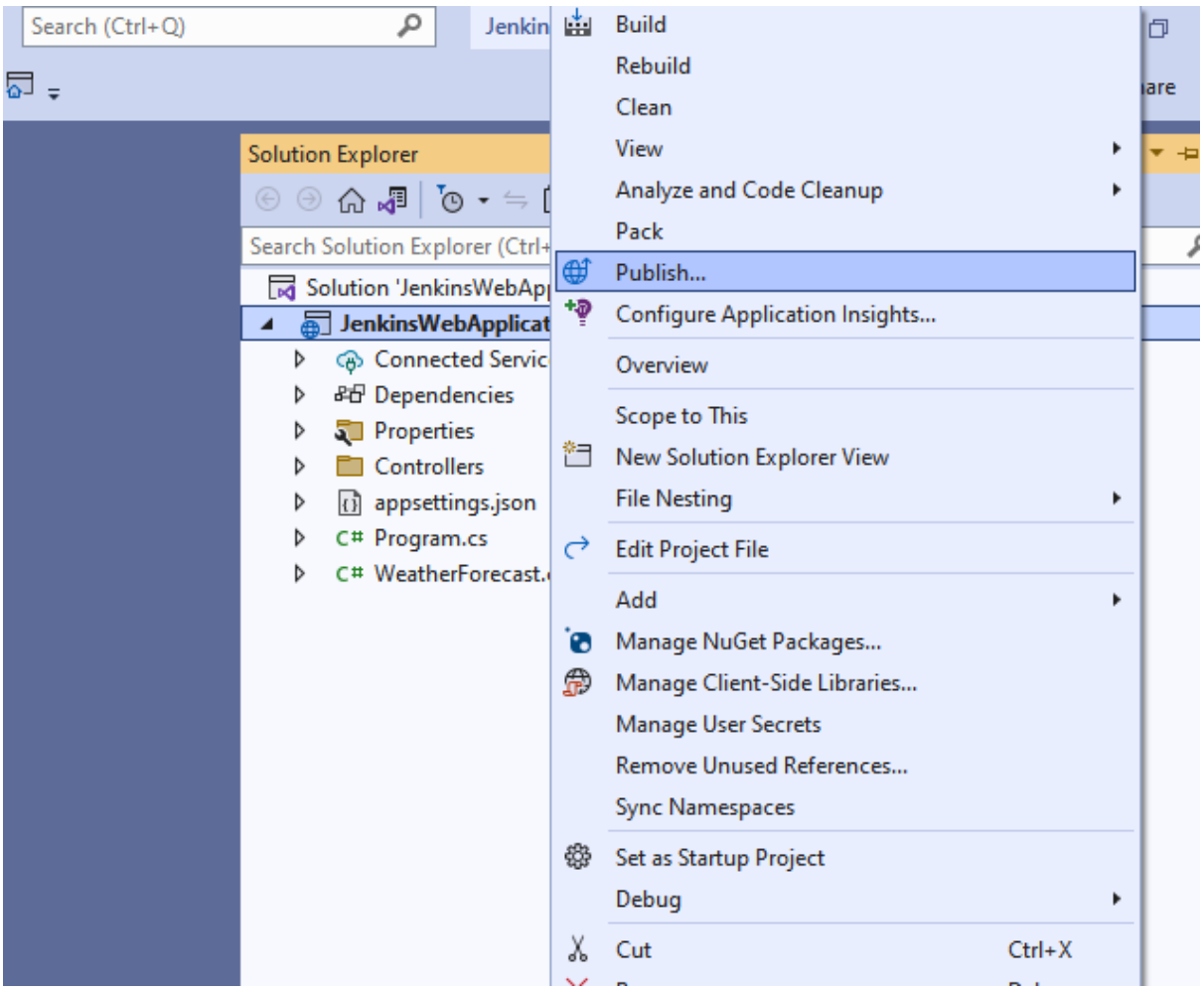
Step 4

Finally, run the application



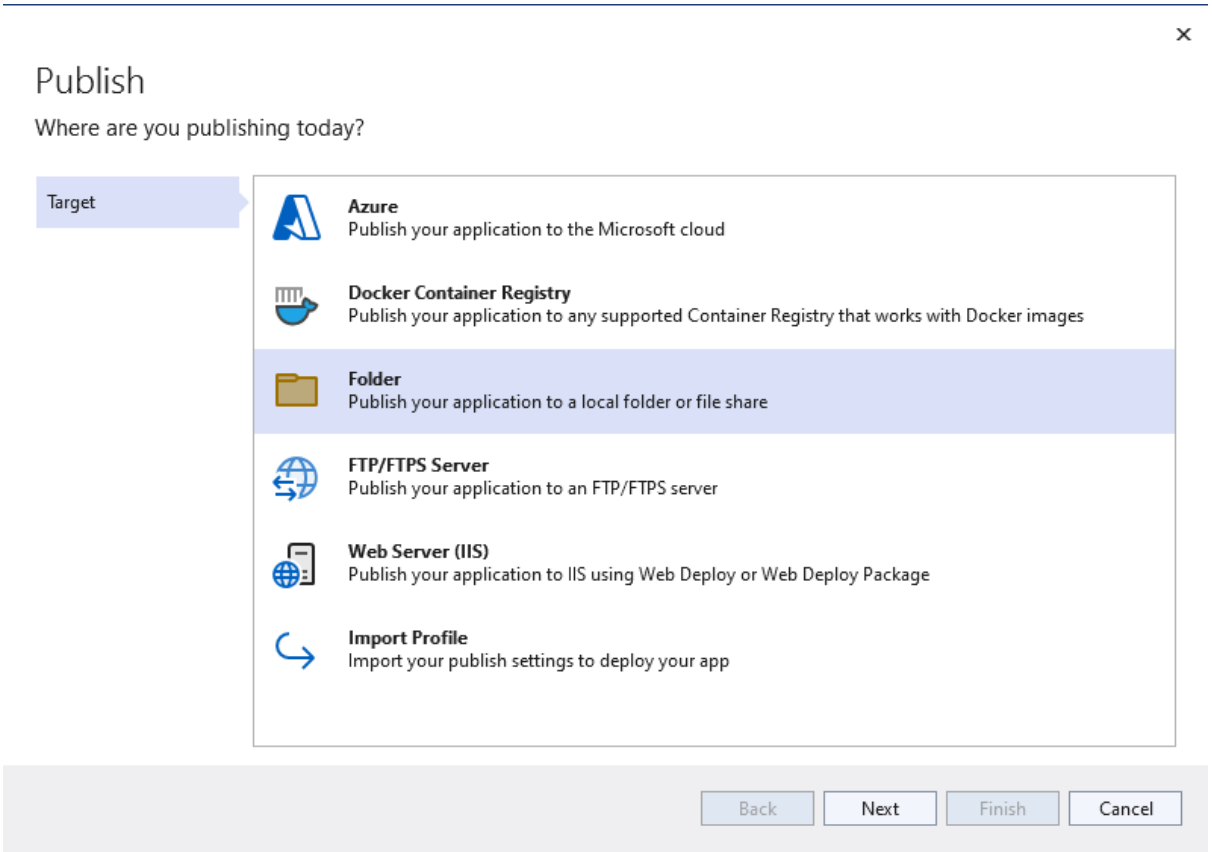
Step 5

Now we are going to publish this code



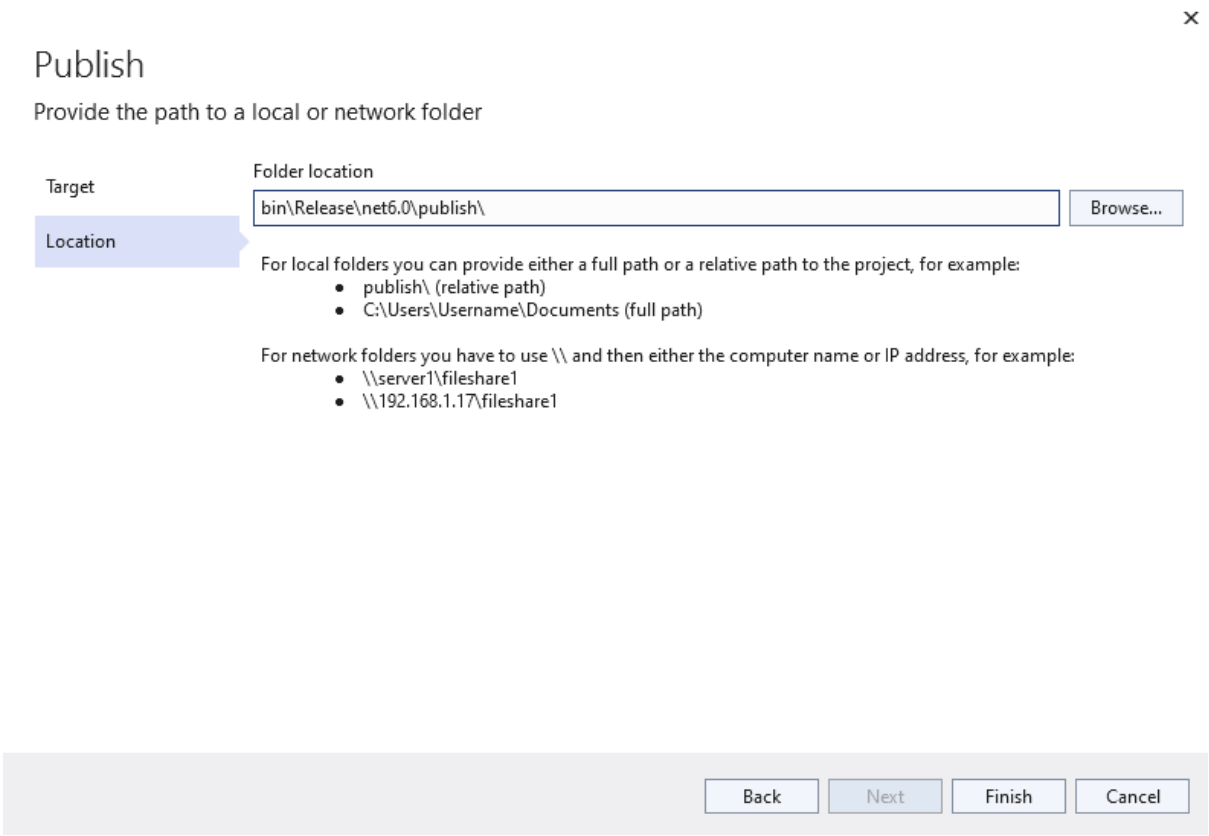
Step 6

Select folder



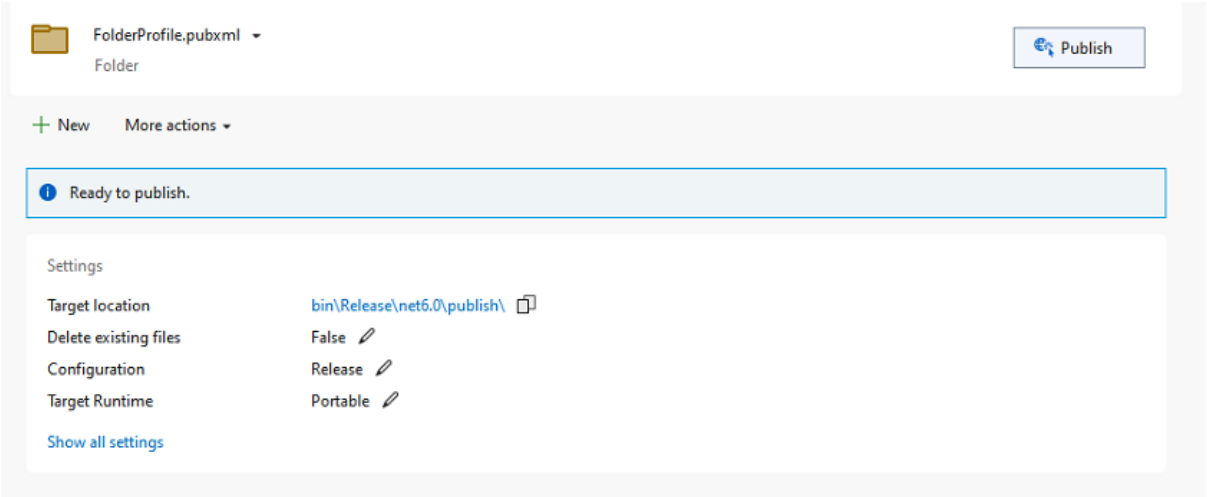
Step 7

It will take default publish path



Step 8

Finally, publish



Step 9

Go to the property section of project inside solution and edit FolderProfile.pubxml file and change the web publish method to Package

```xml
<?xml version="1.0" encoding="utf-8"?>
<!--
https://go.microsoft.com/fwlink/?LinkID=208121.
-->
<Project>
  <PropertyGroup>
    <DeleteExistingFiles>False</DeleteExistingFiles>
    <ExcludeApp_Data>False</ExcludeApp_Data>
    <LaunchSiteAfterPublish>True</LaunchSiteAfterPublish>
    <LastUsedBuildConfiguration>Release</LastUsedBuildConfiguration>
    <LastUsedPlatform>Any CPU</LastUsedPlatform>
    <PublishProvider>FileSystem</PublishProvider>
    <PublishUrl>bin\Release\net6.0\publish\</PublishUrl>
    <WebPublishMethod>Package</WebPublishMethod>
  </PropertyGroup>
</Project>
```

Step 10

We also need Microsoft Web Deploy, and most of time it will come with Visual Studio

Step 11

Later on, we just create an empty xUnit Test Case project just for demo purposes inside the same solution

Step 11

Run your project and confirm all things are working fine on your system before committing to git.

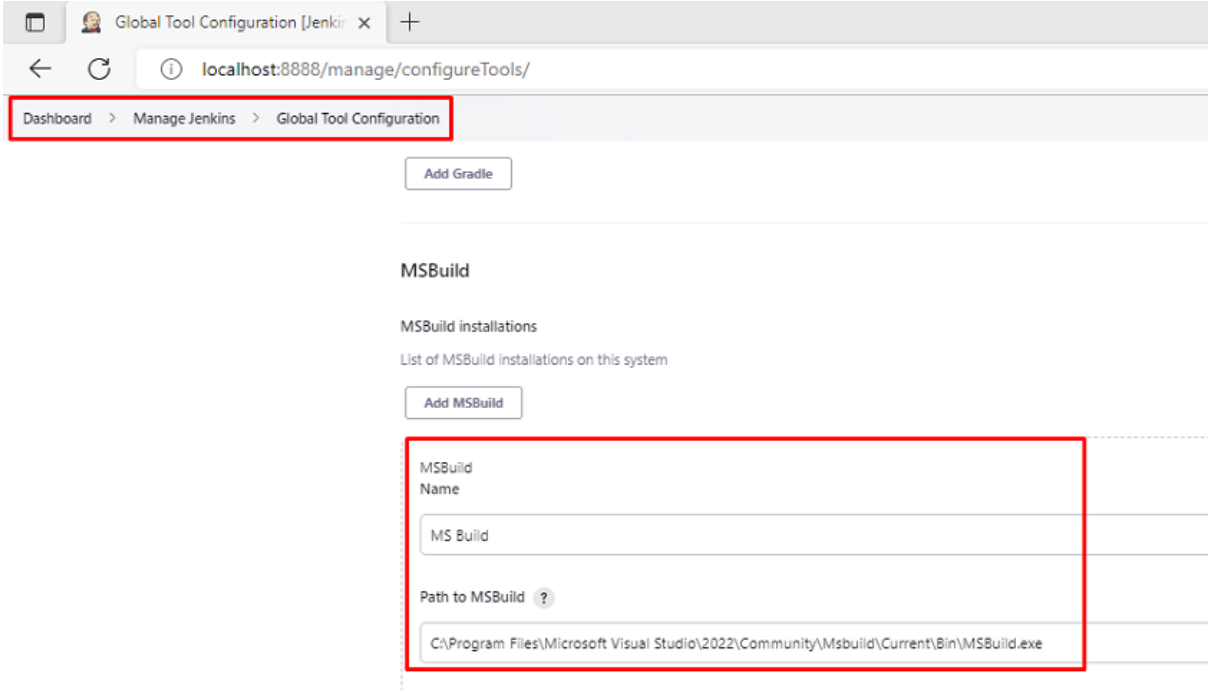
Step 12

Next, create a git repository and push your code into that

Jenkins Pipeline and Configuration

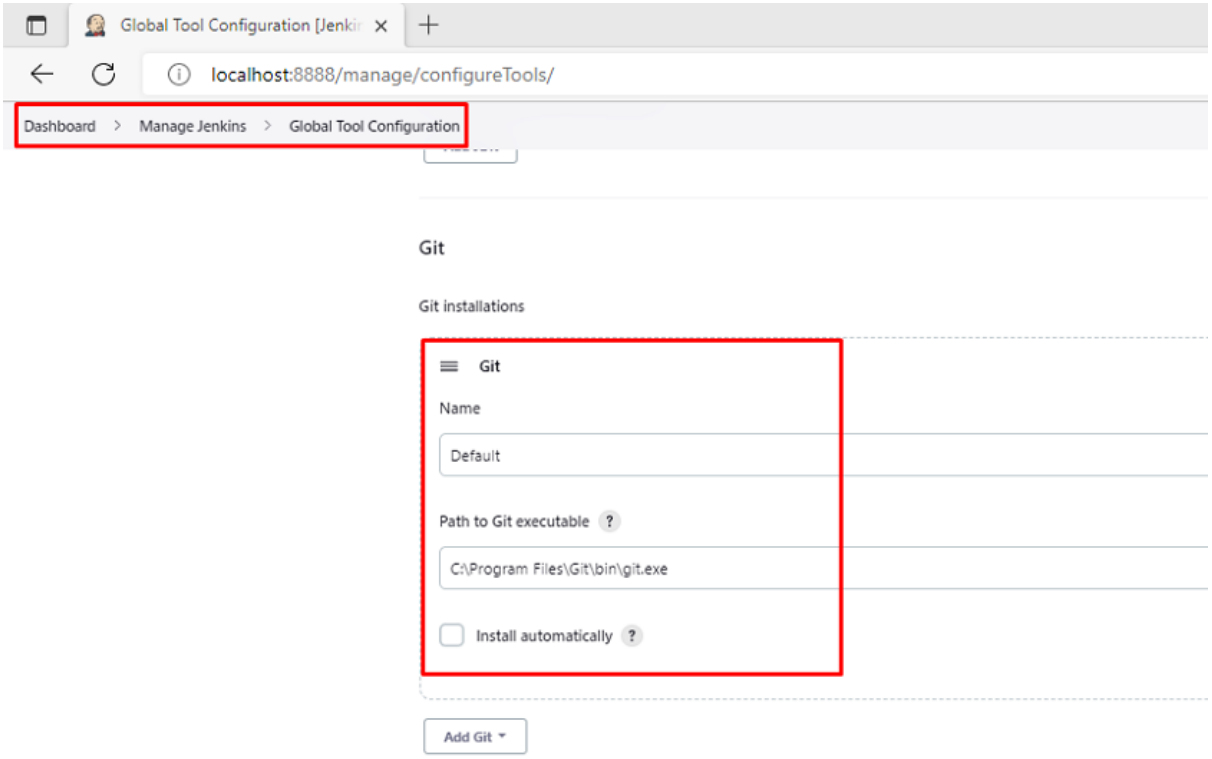
Step 1

First, we are going to add MS Build Path and Configuration for that go to the Global Tool Configuration inside the Manage Jenkins



Step 2

Next, we add the git file path inside the global tool configuration



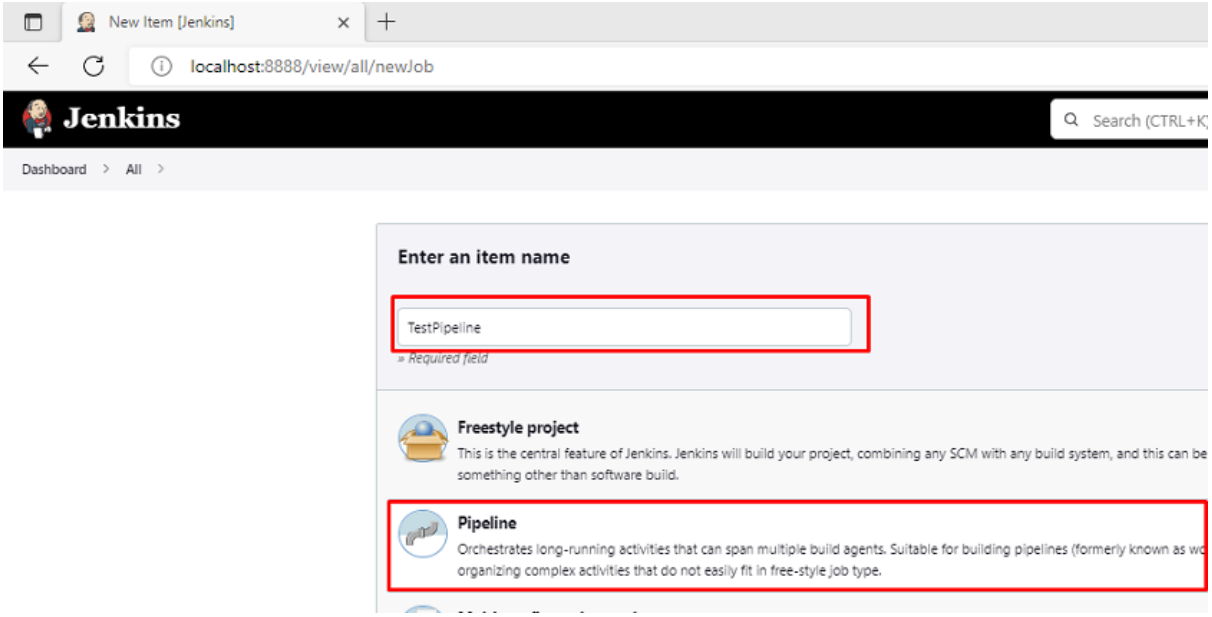
Step 3

Now, we are going to add git credentials in Manage Credentials inside Manage Jenkins click and **global credentials(unrestricted)** add and your git credentials like username and password it will create credential id automatically once you click on save and that id you need to put inside pipeline configuration.

Step 4

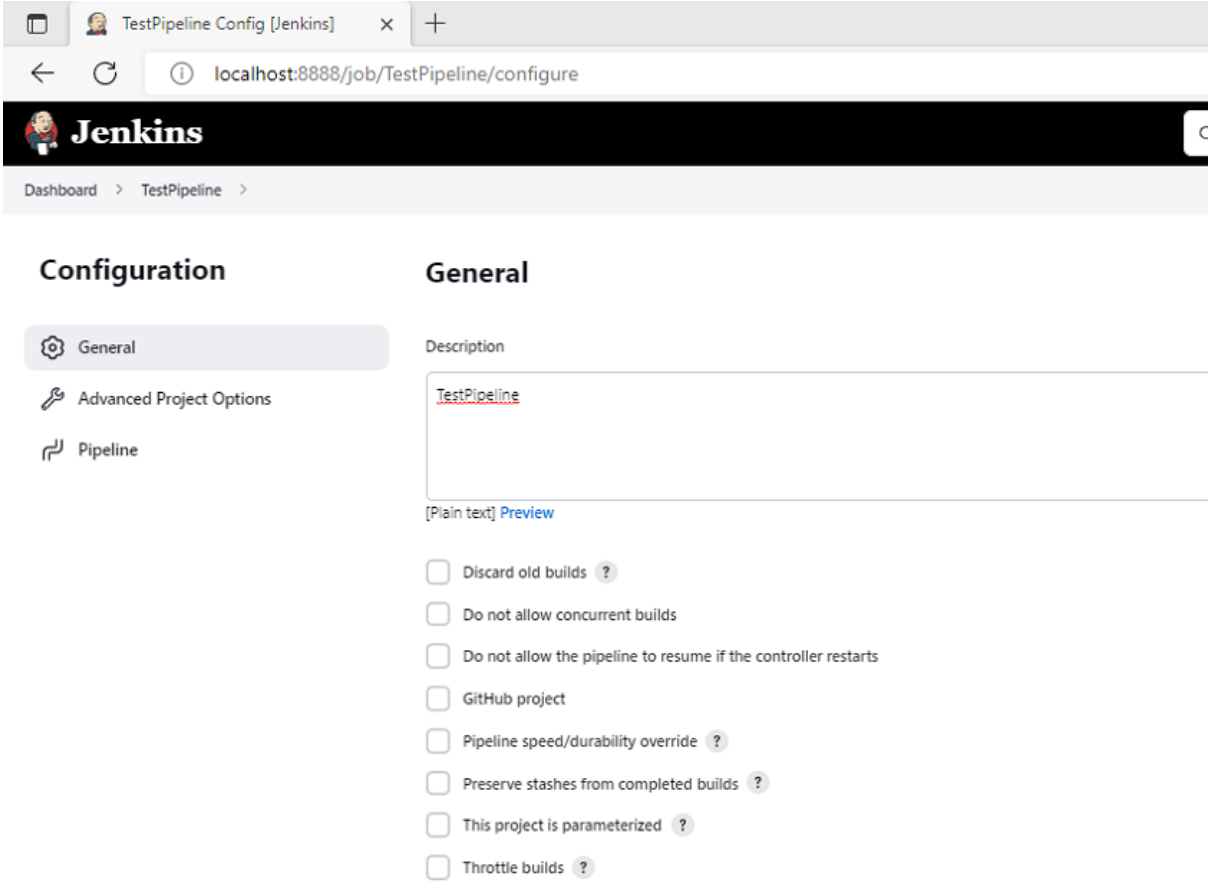
Open Jenkins Dashboard and click on New Item to create a Pipeline

The pipeline is basically the set of steps that are going to execute before creating the build that includes unit test, integration test case, and many more as per need



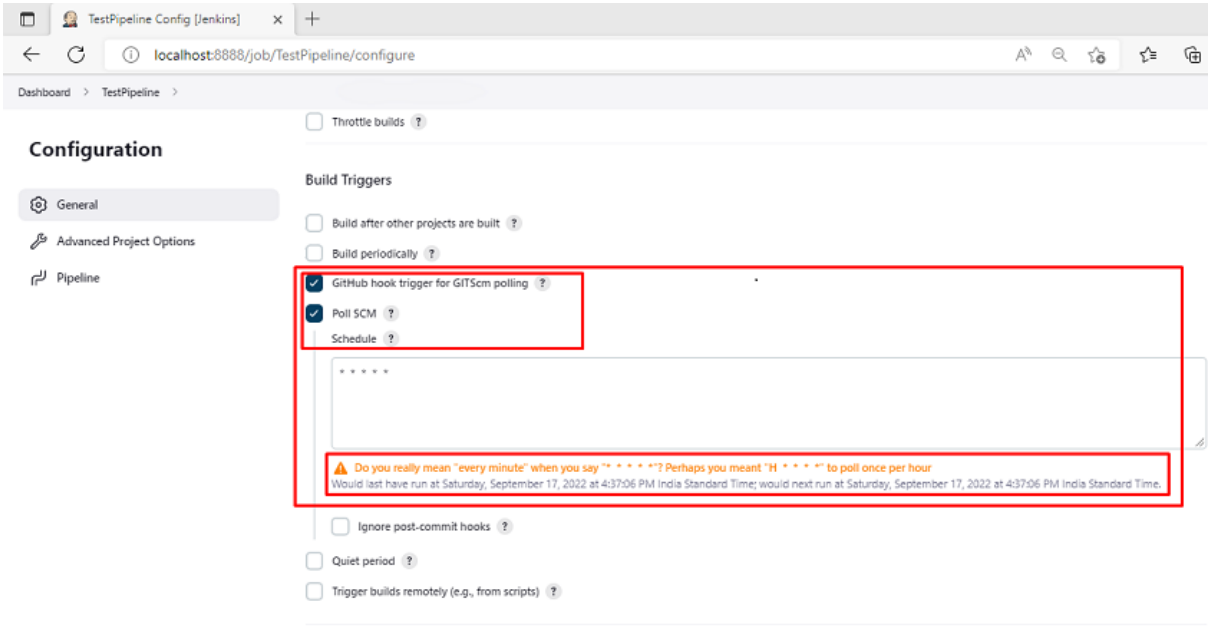
Step 5

Provide general information like a description



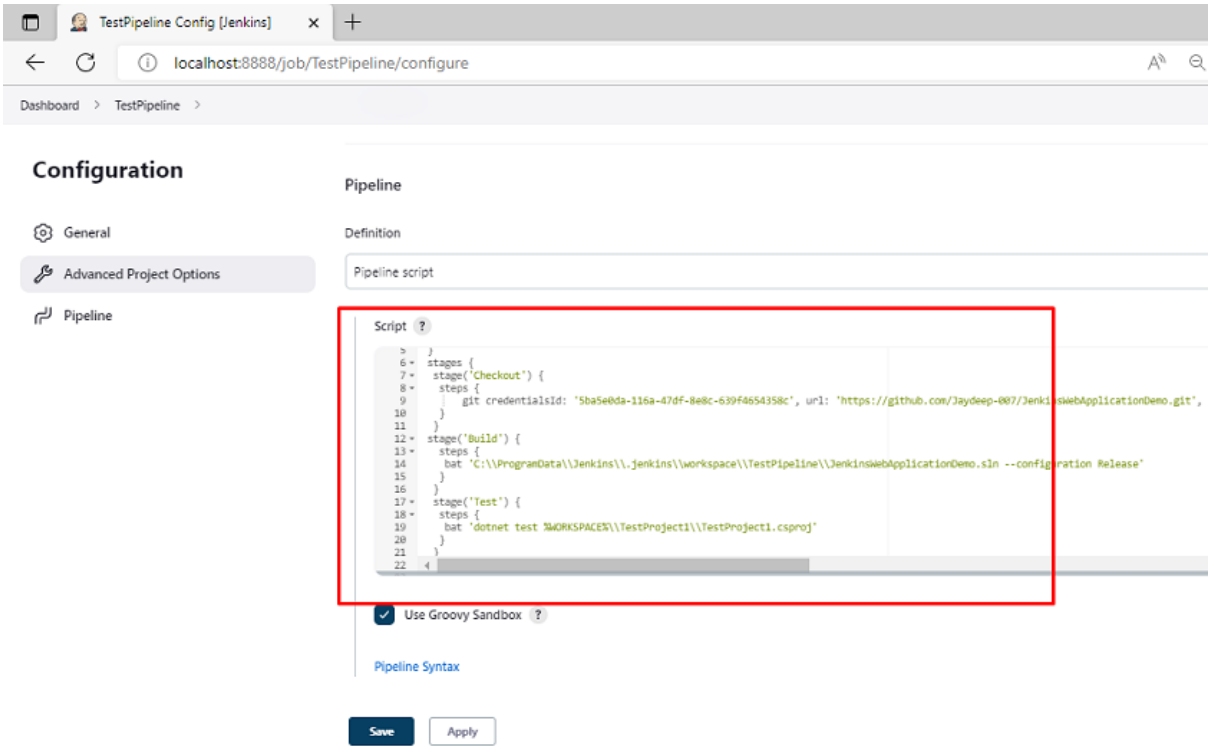
Step 6

Next, click on a few checkboxes which help us to automatically trigger the pipeline whenever we commit code inside git repositories and put five start separated by spaces that is us to trigger pipeline instantly whenever the developers commit the code



Step 7

Add the following pipeline script which is basically a set of steps that are going to executes before creating the build (Note- Please make sure all your paths are corrects otherwise it will throw errors in the build process)

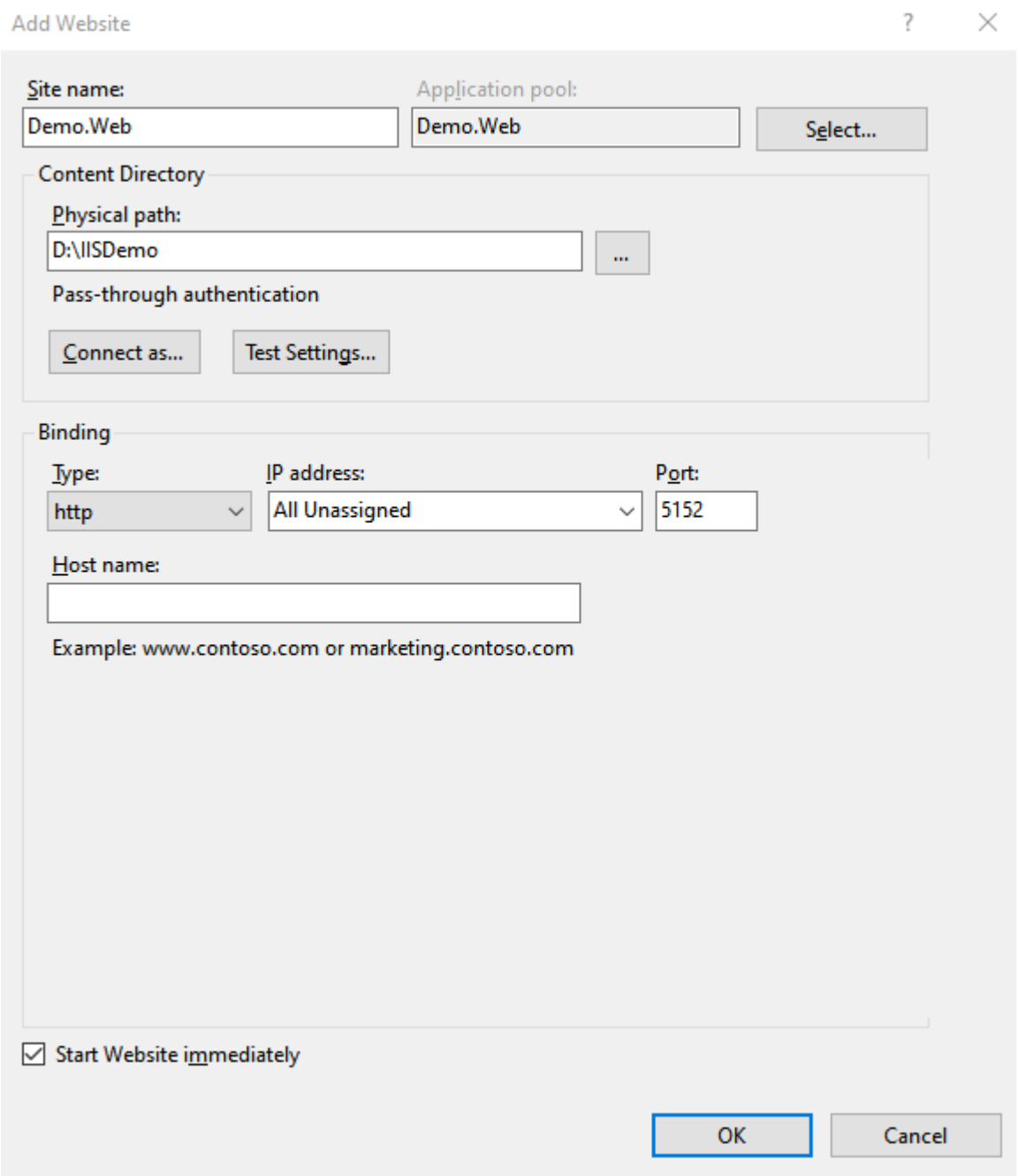


```
pipeline { agent any environment { dotnet = 'C:\\Program Files\\dotnet\\dotnet.exe' } stages { stage('Checkout Stage') { steps { git credentialsId: '5ba5e0da-116a-47df-8e8c-639f4654358c', url: 'https://github.com/Jaydeep-007/JenkinsWebApplicationDemo.git', branch: 'main' } } stage('Build Stage') { steps { bat 'C:\\ProgramData\\Jenkins\\.jenkins\\workspace\\TestPipeline\\JenkinsWebApplicationDemo.sln --configuration Release' } } stage('Test Stage') { steps { bat 'dotnet test %WORKSPACE%\\TestProject1\\TestProject1.csproj' } } stage("Release Stage") { steps { bat 'dotnet build %WORKSPACE%\\JenkinsWebApplicationDemo.sln /p:PublishProfile="%WORKSPACE%\\JenkinsWebApplicationDemo\\Properties\\PublishProfiles\\FolderProfile.pubxml" /p:Platform="Any CPU" /p:DeployOnBuild=true /m' } } stage('Deploy Stage') { steps { //Deploy application on IIS bat 'net stop "w3svc"' bat '"C:\\Program Files (x86)\\IIS\\Microsoft Web Deploy V3\\msdeploy.exe" -verb:sync -source:package="%WORKSPACE%\\JenkinsWebApplicationDemo\\bin\\Debug\\net6.0\\JenkinsWebApplicationDemo.zip" -dest:auto -setParam:"IIS Web Application Name"="Demo.Web" -skip:objectName=filePath,absolutePath=".\\\\PackagDemoeTmp\\\\Web.config$" -enableRule:DoNotDelete -allowUntrusted=true' bat 'net start "w3svc"' } } } }
```

JavaScriptCopy

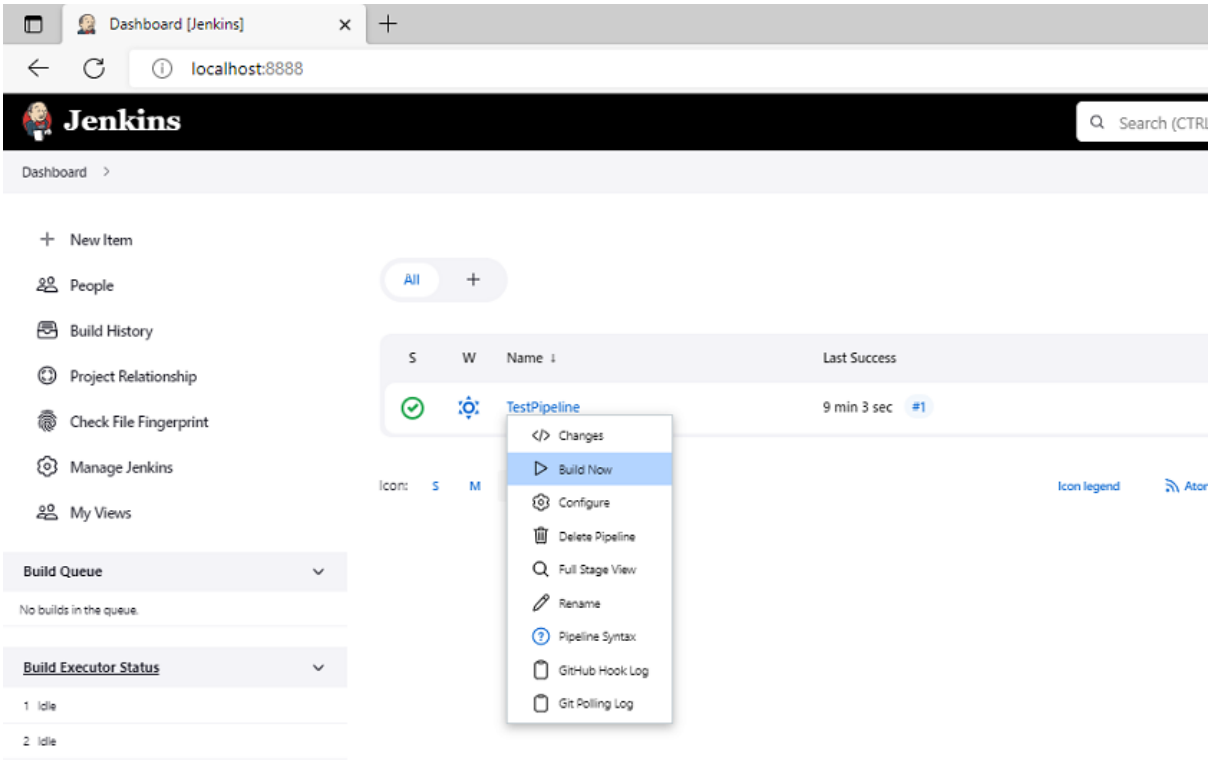
Step 8

Open the IIS and create empty website and point to the any empty folder in that our pipeline will publish and deploy our code after build is getting completed



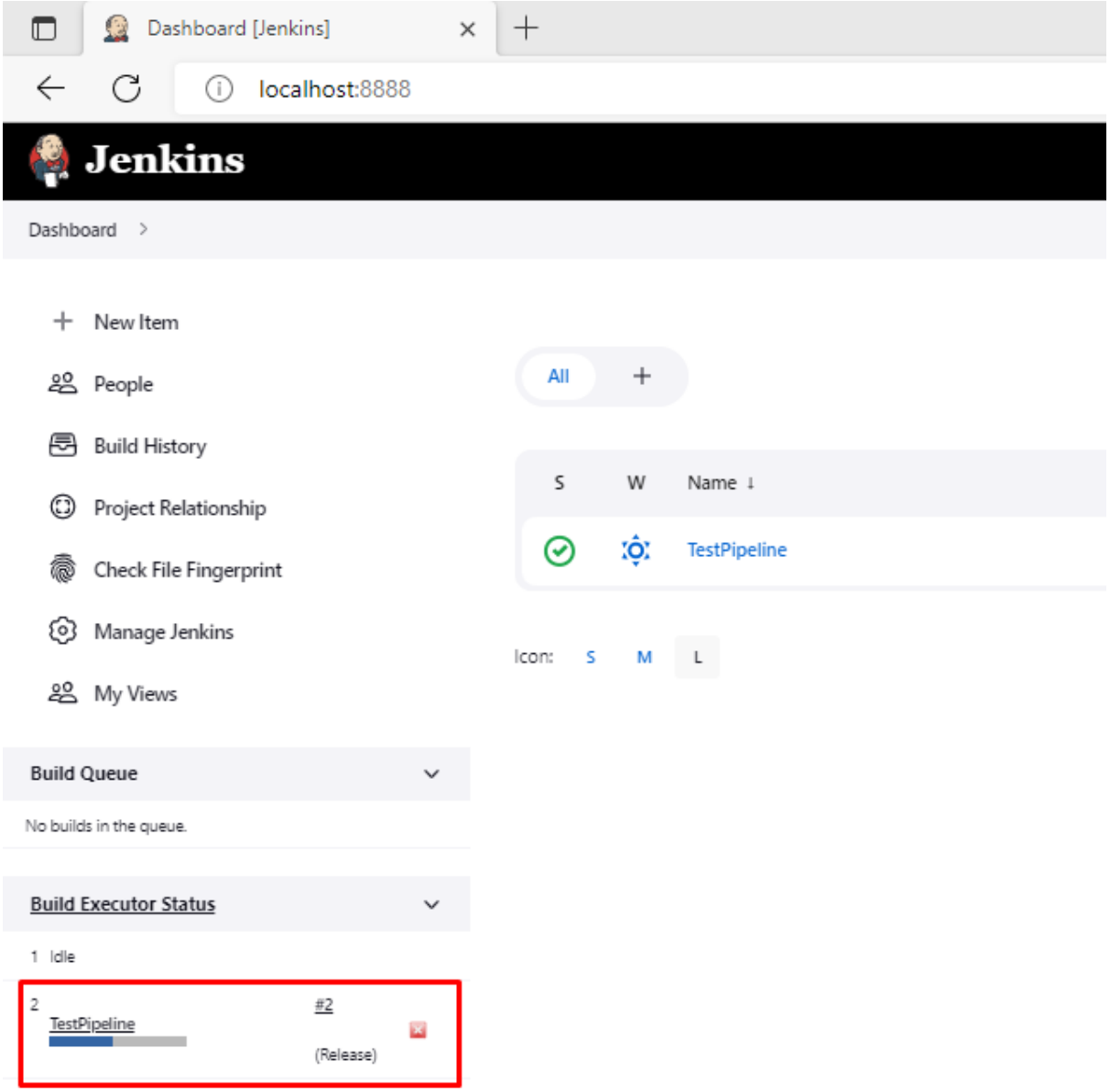
Step 9

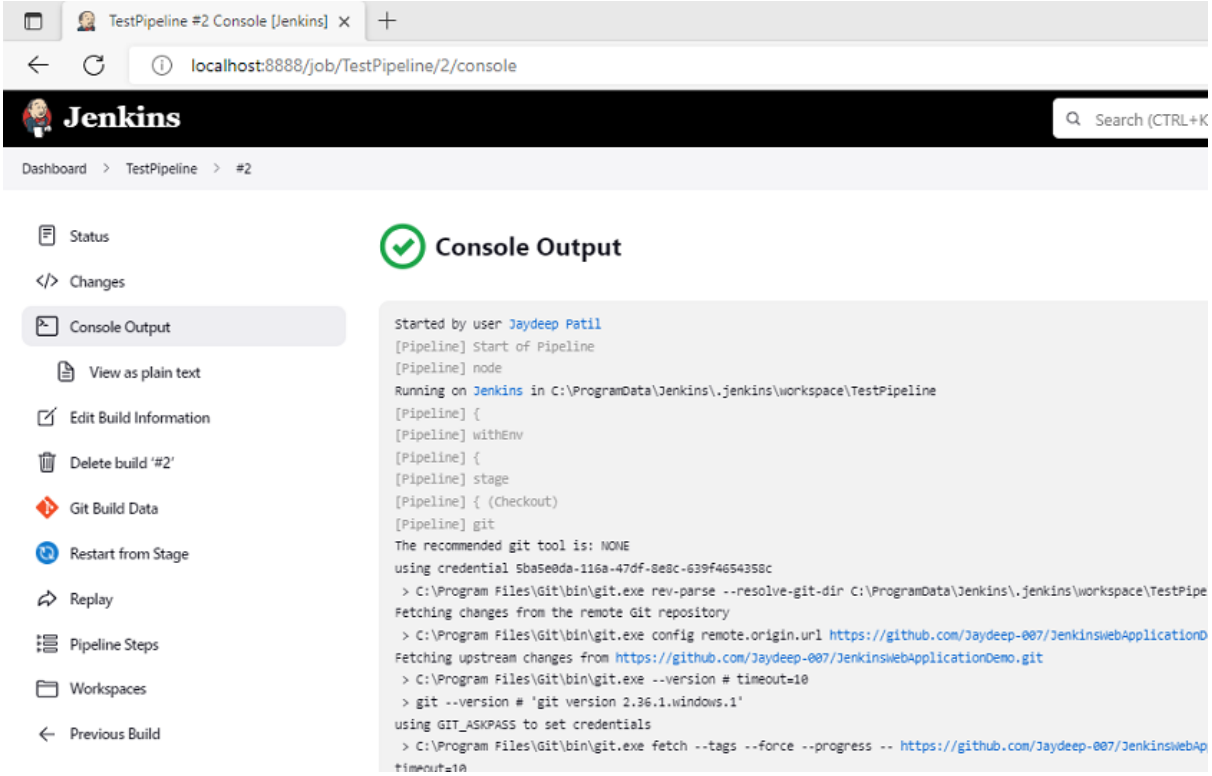
After that go to the dashboard and run your pipeline

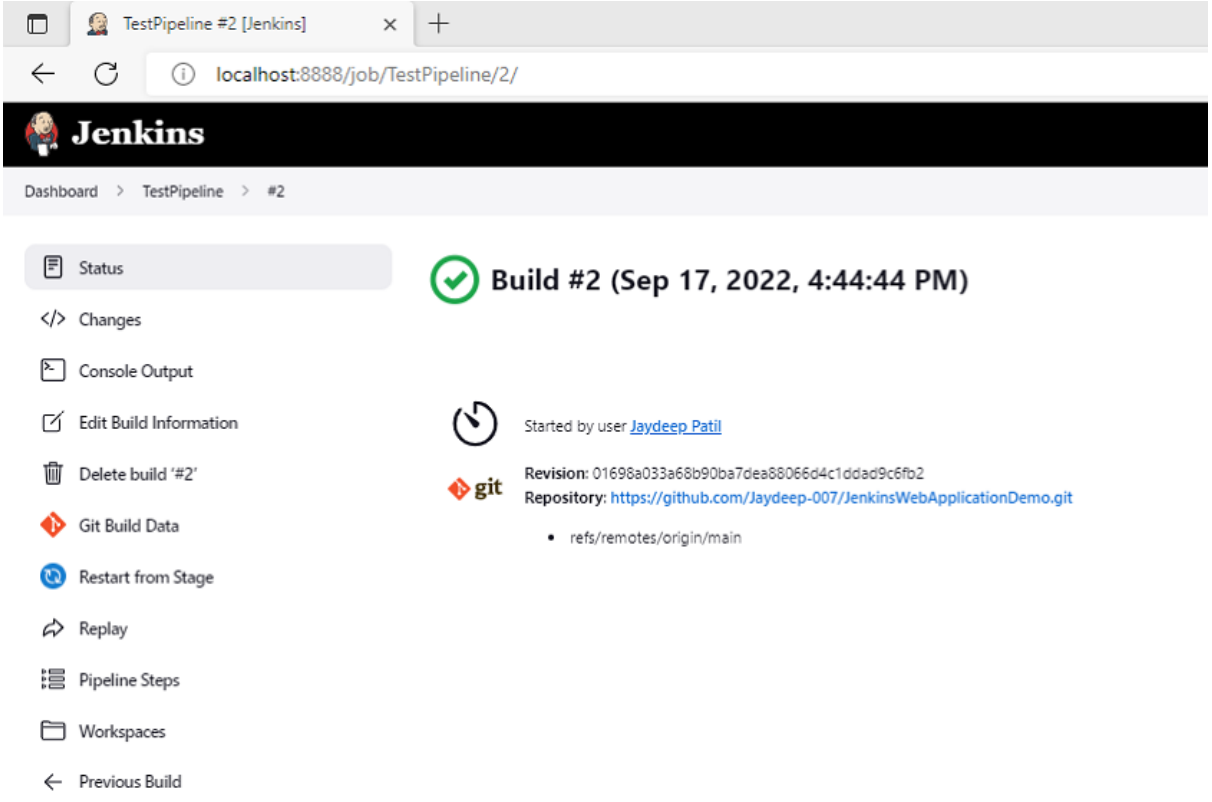


Step 10

Here you can see the build is started


Dashboard [Jenkins]
localhost:8888
Jenkins
Dashboard
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2
TestPipeline
#2
(Release)
All
S
W
Name
TestPipeline
Icon:
S
M
L





Now whenever you change add commit the code then build is triggered and the pipeline is going to create a new build for us and deploy it inside the IIS. This process is continuously running. Also, if the build failed due to some test cases and

something like that then you can also configure the email using SMTP protocol inside Jenkins that helps us build

Step 11

Finally, Open IIS and you can now access your application with the latest build

Step 12

Browse the URL from IIS and you can see the application is in running mode