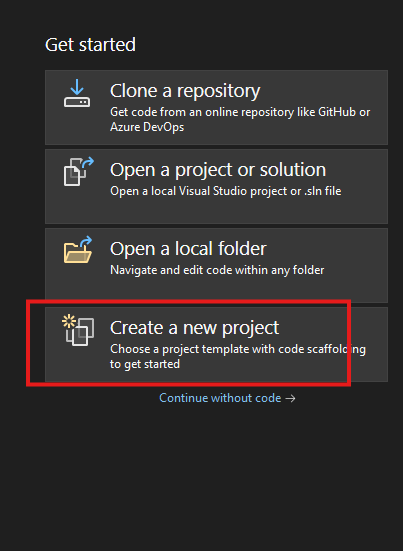
Setup Project on Sonar Cloud

### Setup Project and Write Test Case

1. Create new .net project or you can use your existing also

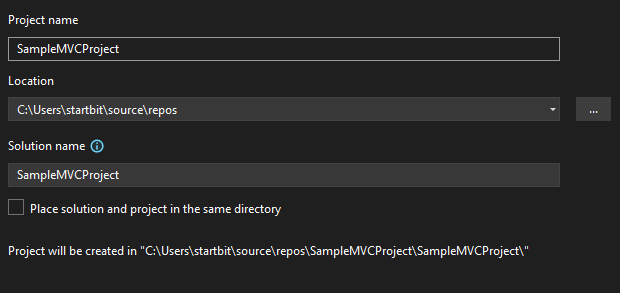
* Click on Create a new Project



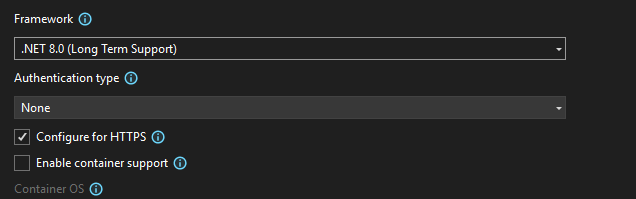
* Select project type, I select MVC, you can choose as per your requirement



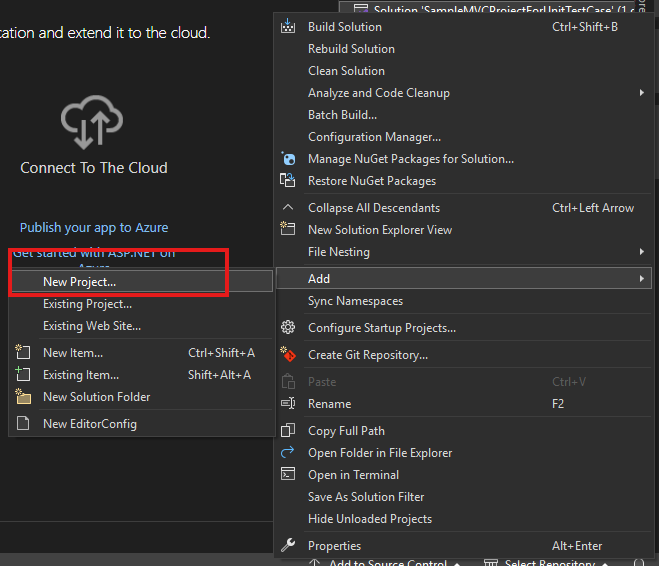
* Give name to your project



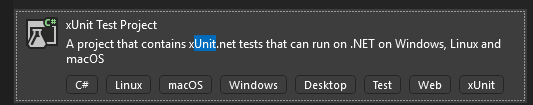
* Select .net version



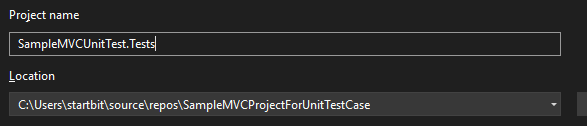
* Click on Create button, and your project is ready.
* For Create Unit Test Case Project.
* Go on Solution of your project, click on Add => New Project



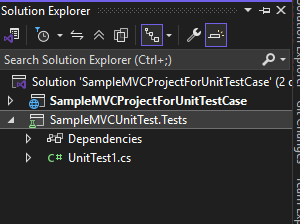
* Select “xUnitTestProject”



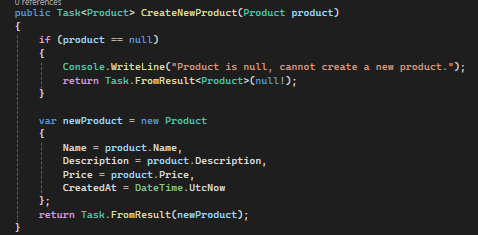
* Give Project name like this, project name should contains .Tests in the end



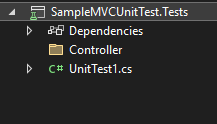
* You can see in the solution, Unit test case project is created



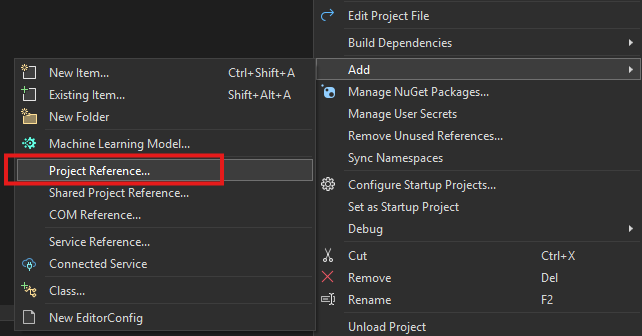
* Let’s create a new Method and create a new test case for that method.
* Let’s create a new method in HomeController.cs



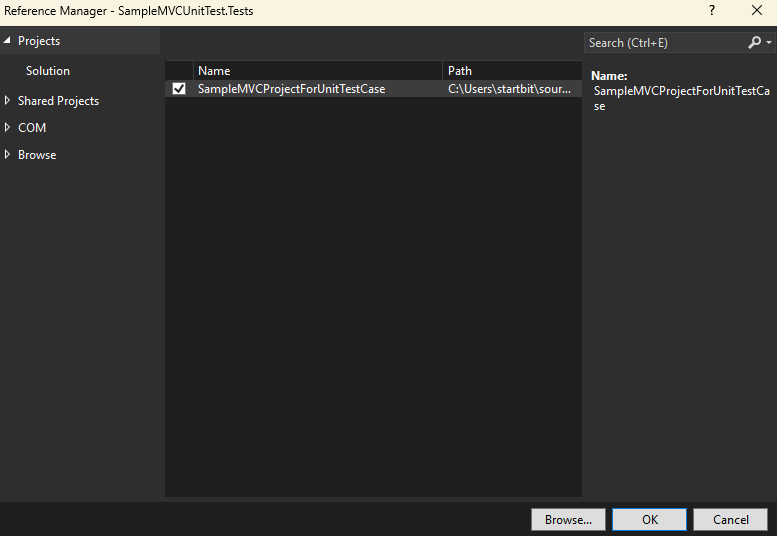
* Now create a new Folder in Unit Test Case project, “Controller”



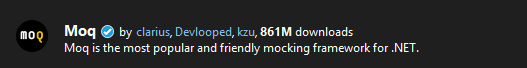
* Now in the controller, create a new file name HomeControllerTest.cs
* Give project reference unit test case to Main project.



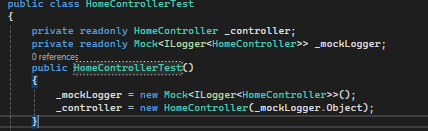
* Select the project and click on Ok button.



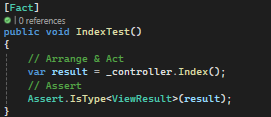
* Add nuget package in Unit Test case project



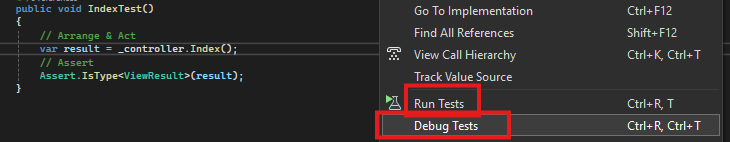
* Project is all set up now we can write unit test case for method.
* Setup Test controller like this



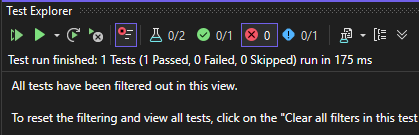
* There are three type of test cases
* 1ST [Fact]
* Let’s write test case for Index method, which is in HomeController.cs



* In the Arrange part, if we have many dependencies in the controller, we can arrange required data through mock in this part, but in this method we don’t have any required decencies so we can direct execute the method.
* In the Act part, we execute the method.
* In the assert part, we match the result, like it’s returning correct result type or not.
* For run the unit test case, right click in the test area =>
* For direct run click on Run Tests
* For debug the test click on Debug Tests

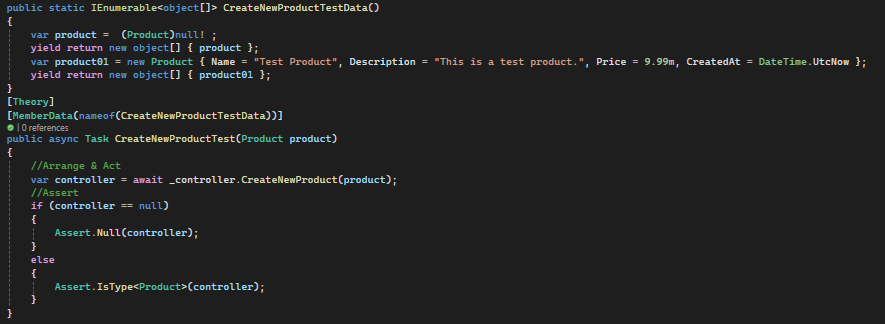


* When we click on debug test, debugger hits our method. In the Test Explorer we can check our test result, it failed or passed.

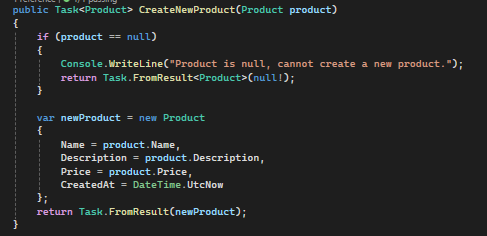


Note: Generally, we don’t use of Fact type, we only use Fact type test case when we have small method and not nested condition.

* 2nd type is Theory type
* This type is use for multiple nested condition and multiple condition check.

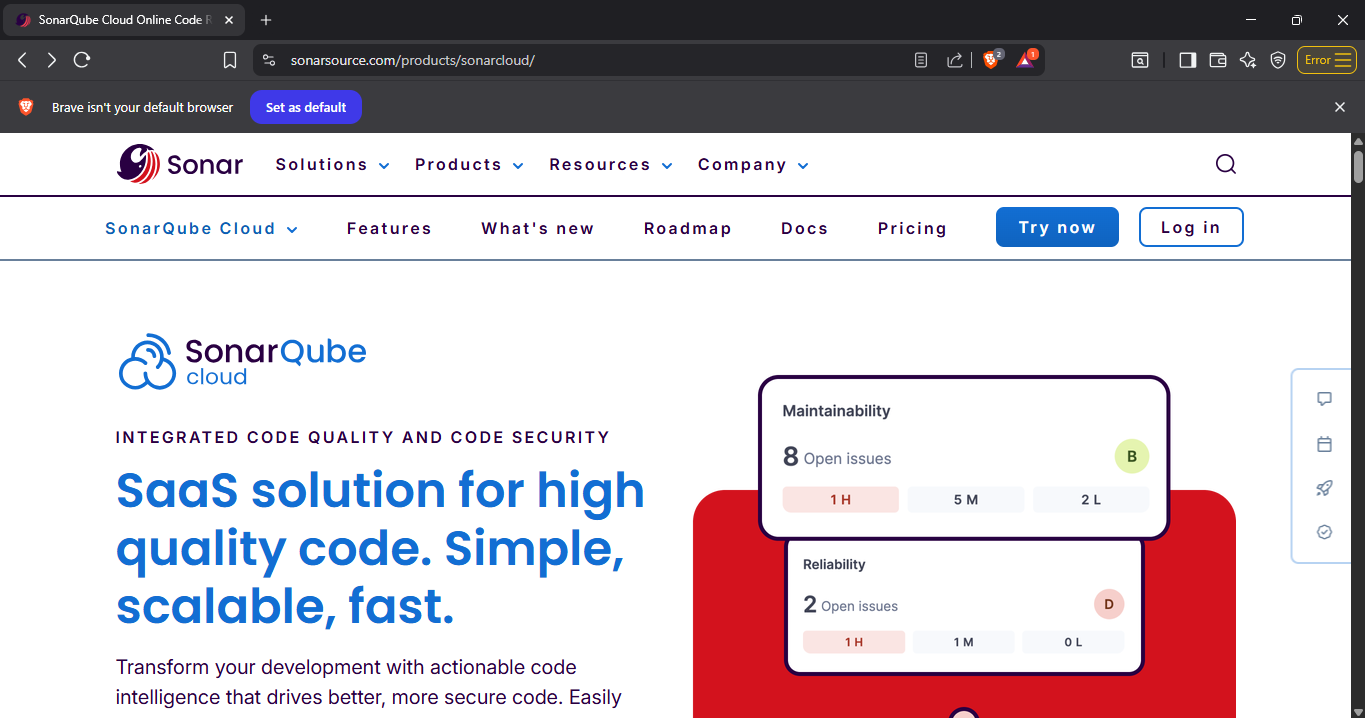


* Let’s breck this in a part
* 1st part is Upper section TestData (“CreateNewProductTestData”), In this part we set the dummy value for method.
* 2nd part TestMethod(CreateNewProductTest)
* In this Arrange all the required data and pass to a method a method and check the result type.
* We created two scenario and data according to scenario because we have 2 condition in the method for check. One is when product is null or second is when product has data.

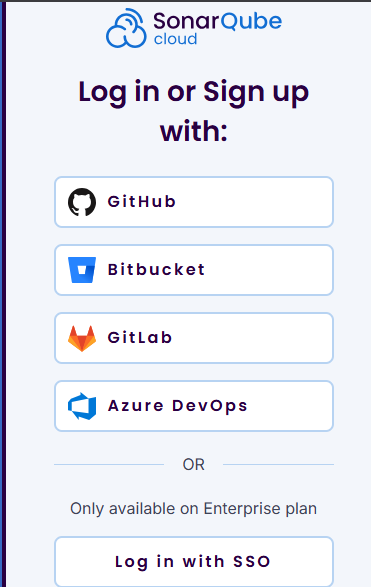


### Setup Project on Sonar Cloud

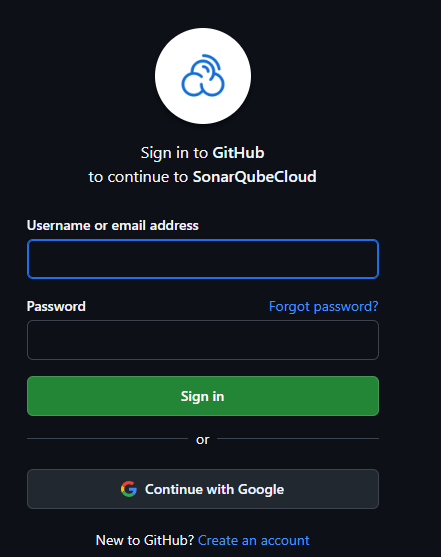
* Go in SonarCloud.io



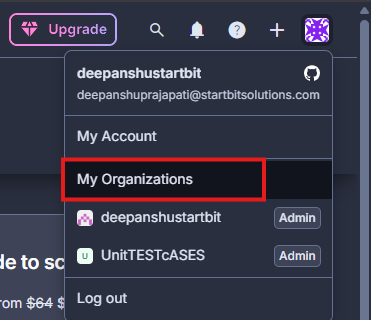
* Click on Log In Button, then choose as per requirement, I am going with GitHub because my project is on Git.



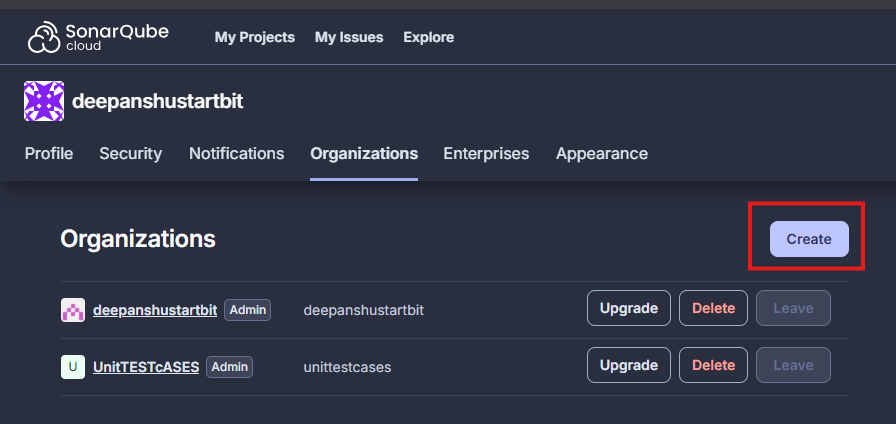
* Fill your details & click on sign In button



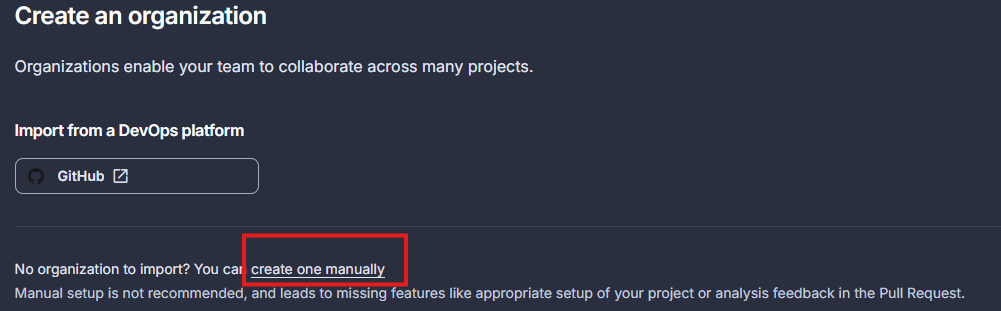
* Click on the My Organization button



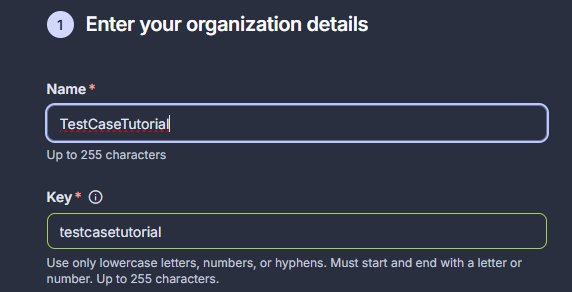
* Click on Create button



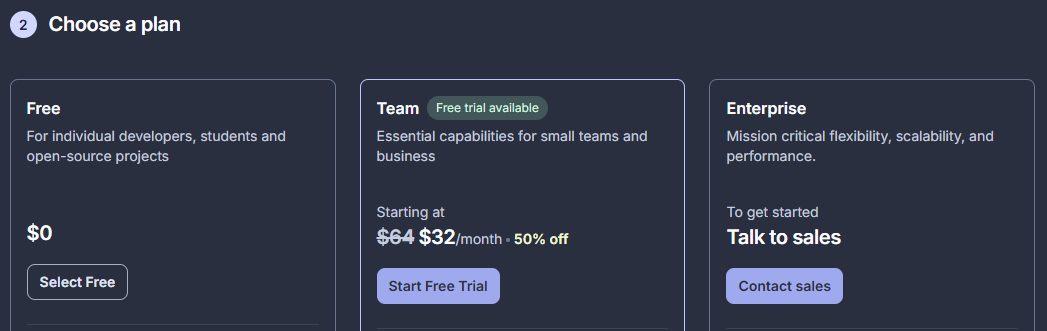
* Click on GitHub or Create one Manually, I am going with manually because I have already 2 organization



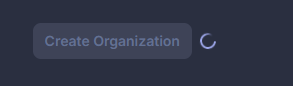
* Fill the details



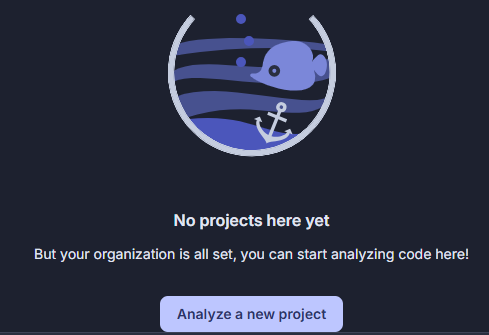
* Choose your plan, I am going with free trial



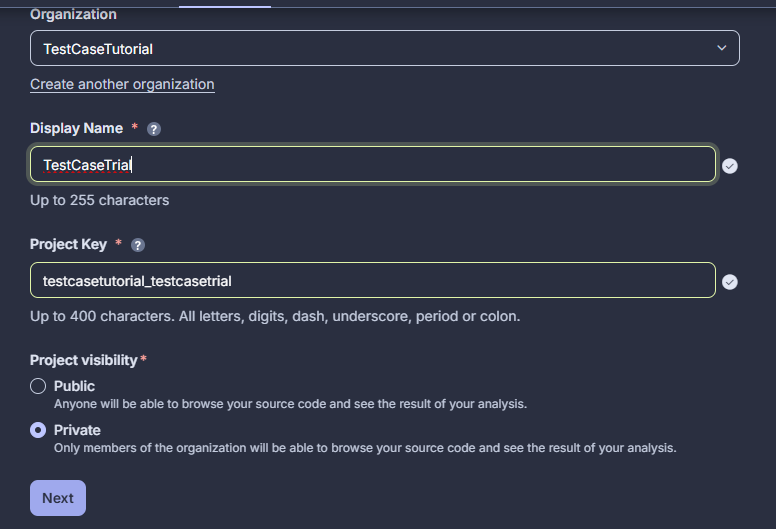
* Click on Create a Organization button



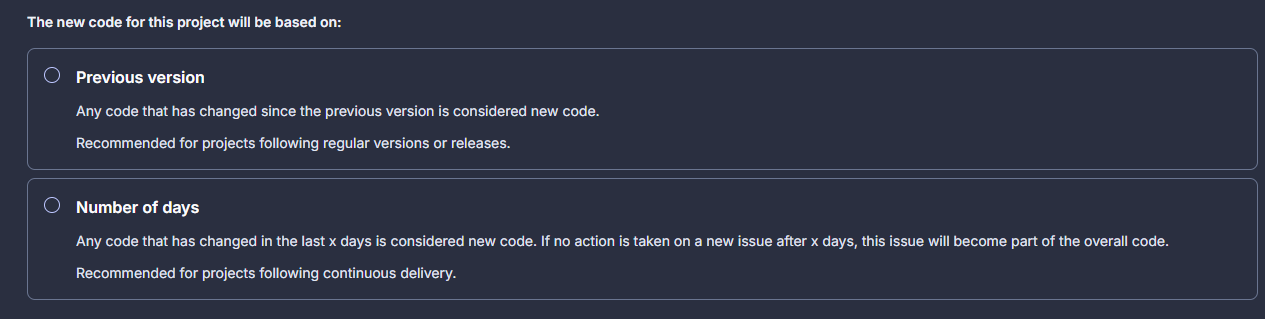
* Scroll down and click on Analyze a new project



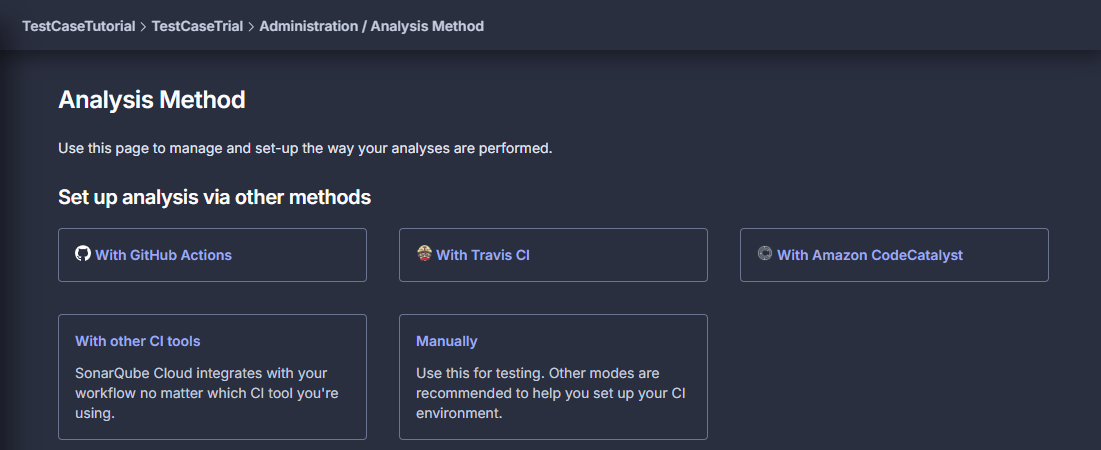
* Fill the project display name and click on the Next button



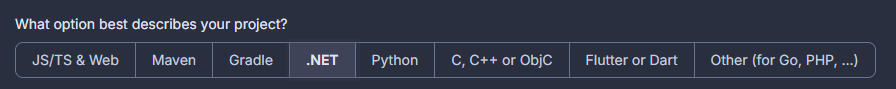
* Select new code conditions, I selected “Previous Version” and click on Create Project button.



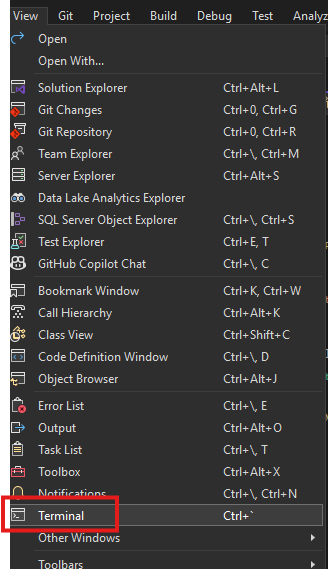
* Select your analysis method



* 1st Click on Manually, we run the report through vs code
* Select



* Go on project in vs code in click on View and Terminal



* Install coverlet.msbuild(6.0.4)
* For generate Coverage file run this command in the powersheel

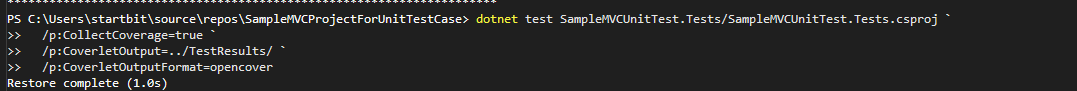
dotnet test SimpleMVCTestApp.Tests/SimpleMVCTestApp.Tests.csproj `

/p:CollectCoverage=true `

/p:CoverletOutput=../TestResults/ `

/p:CoverletOutputFormat=opencover

* Change you project name first in the command



* Run this command and change k(project name) and o(organization name) property to the sonar property which showing in the your account



dotnet sonarscanner begin `

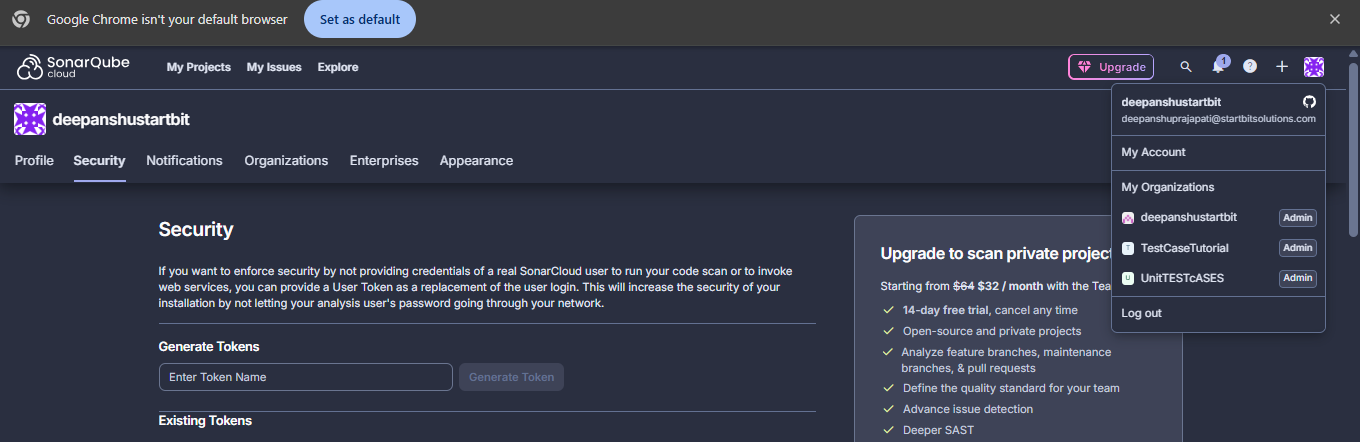
/k:" testcasetutorial\_testcasetrial" `

/o:"testcasetutorial" `

/d:sonar.login="08ba26040c0ed2334e1be43a778281df5c9116bb" ` (token)

/d:sonar.cs.opencover.reportsPaths="TestResults/coverage.opencover.xml"

* For generate Token click on My Organization and generate Token



* Then run this command

dotnet build

* Then re run this command

dotnet test SampleMVCUnitTest.Tests/SampleMVCUnitTest.Tests.csproj `

/p:CollectCoverage=true `

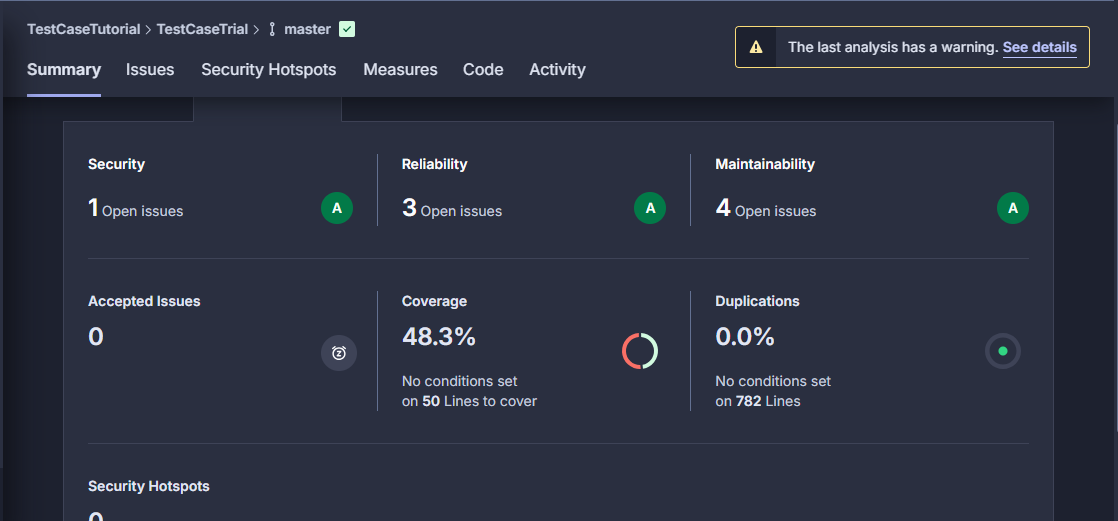
/p:CoverletOutput=../TestResults/ `

/p:CoverletOutputFormat=opencover

* Than run final command

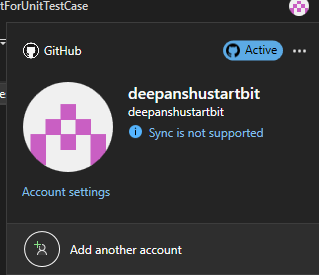
dotnet sonarscanner end /d:sonar.login="eb2b73ed757086b3626a6d07fa38fde9e726509e"

* Refresh the sonarcloud and go your project
* It’s showing the coverage report now

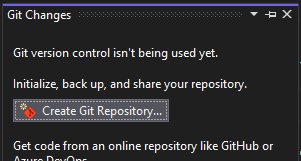


## Push Project Code on GiHub

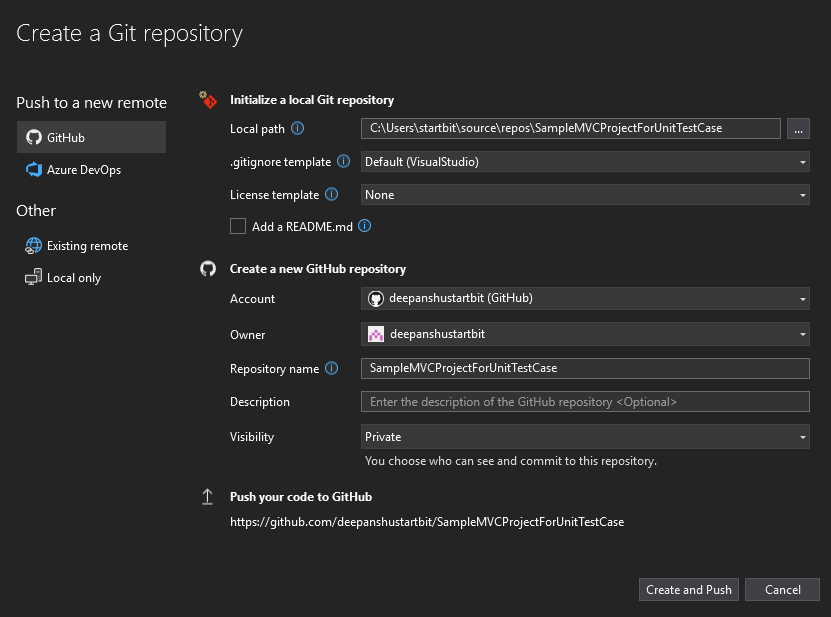
* Log in Git account in vs



* Click on Create Git Repository



* Give repository name and click on Create and Push.



* It’s showing in Git

