



# Publish Code Coverage to SonarCloud Using Coverlet

---

.NET CORE

*Based on a  
presentation  
by John Kear.*

[HTTPS://GITHUB.COM/COVERLET-COVERAGE/COVERLET/BLOB/MASTER/DOCUMENTATION/VSTESTINTEGRATION.MD](https://github.com/coverlet-coverage/coverlet/blob/master/documentation/vstestintegration.md)

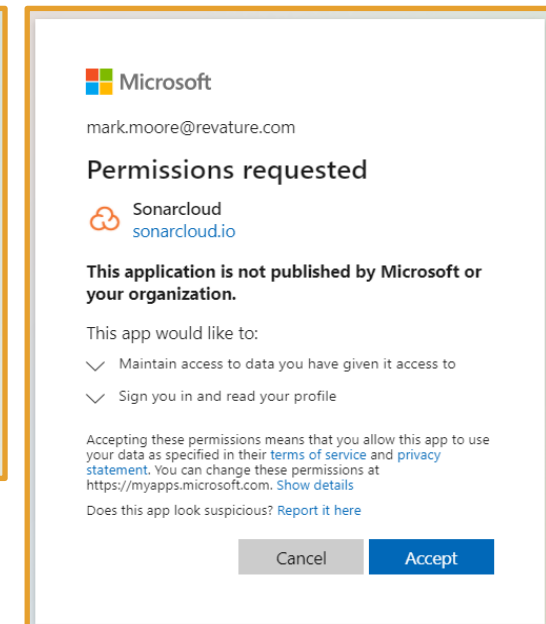
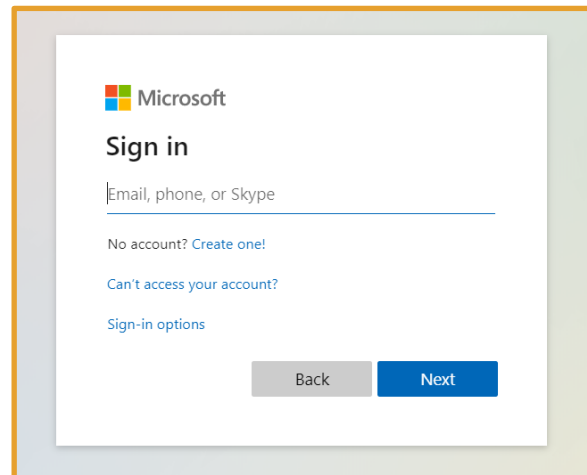
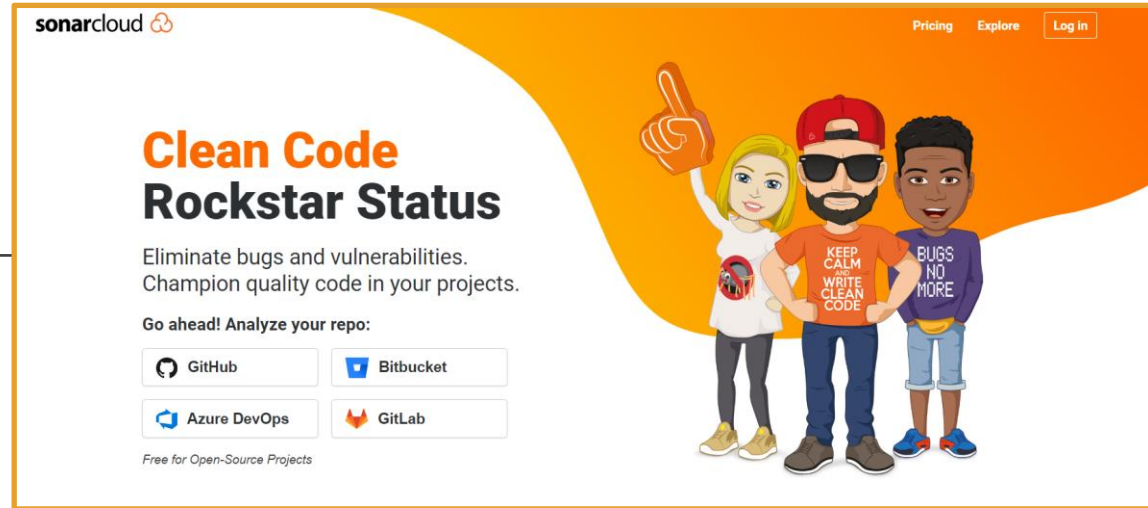
# First Steps

<https://sonarcloud.io/>

First, create a pipeline that successfully builds and deploys to your Website.

Next, log in to [www.sonarcloud.io](https://sonarcloud.io) using your Azure DevOps credentials.

If asked, accept any request for permission or access by Sonarcloud.



# Create SonarCloud Organization

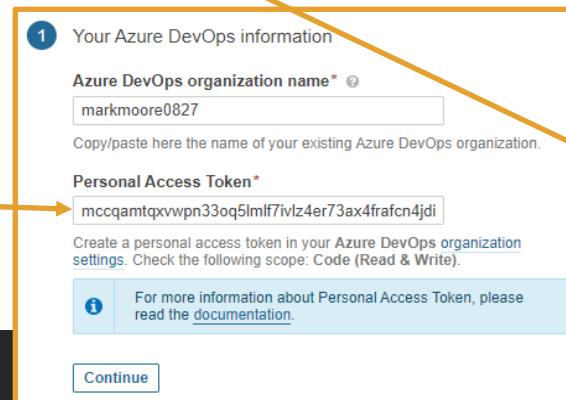
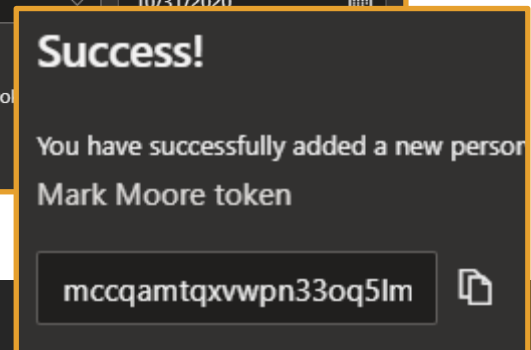
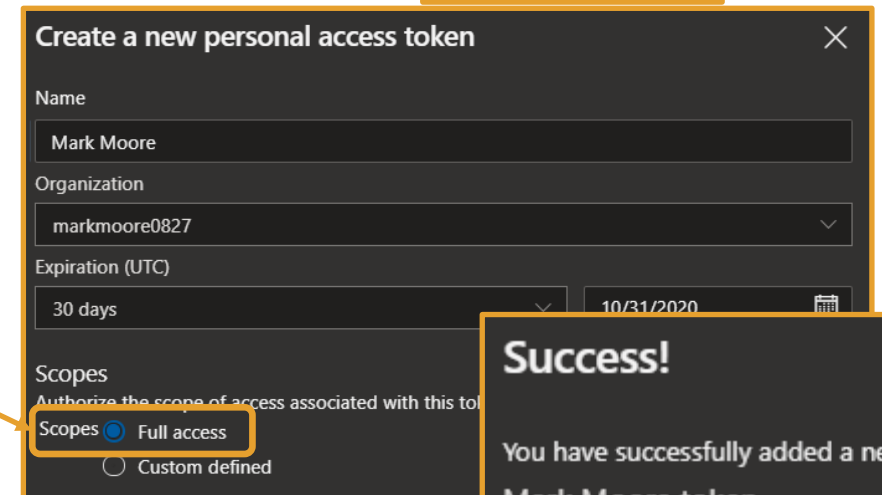
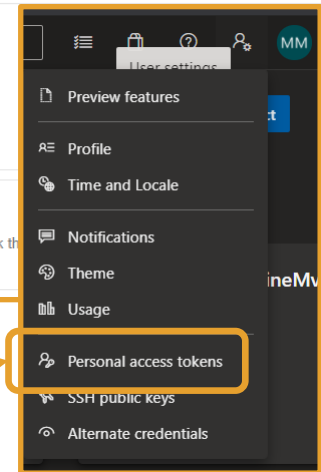
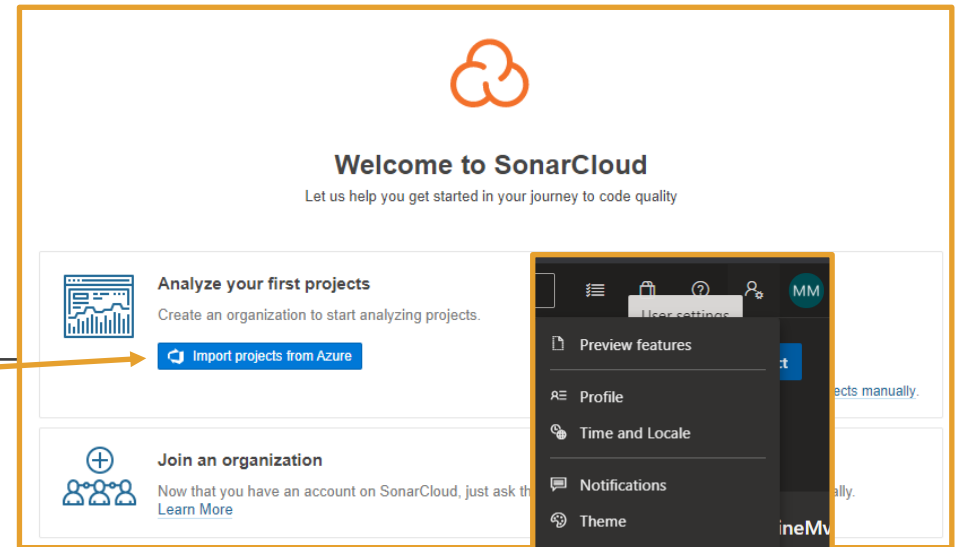
1. Create an organization
2. Enter the correct name of your Azure DevOps Organization and a Personal Access Token (P.A.T.).

1. Log into your Azure Account.
2. Click on User Settings>Personal Access tokens.

3. Click 'New Token'.
4. Enter your name and select 'Full Access'

5. Select 'Create'
6. Copy your P.A.T.

3. Enter the P.A.T.
4. Click 'Continue'



# Create SonarCloud Organization


---

1. Click 'Continue' to  
Import your  
organization details

2. Select 'Free plan'

3. Click "Create  
Organization"

2 Import organization details

Import  markmoore0827 into a SonarCloud organization

**Key\***

Up to 255 characters. Only lower-case letters (a to z), digits or dashes are permitted. A dash cannot occur as a leading or trailing character.

[Add additional info](#) ▼

[Continue](#)

3 Choose a plan

☒ Free plan €0

All projects you analyze will be public.  
Anyone can browse source code.

☐ Paid plan

- ✓ Unlimited private projects
- ✓ Strict control over visibility
- ✓ No commitments, cancel anytime
- ✓ 14 days free trial.

[Learn more](#)

[Create Organization](#)

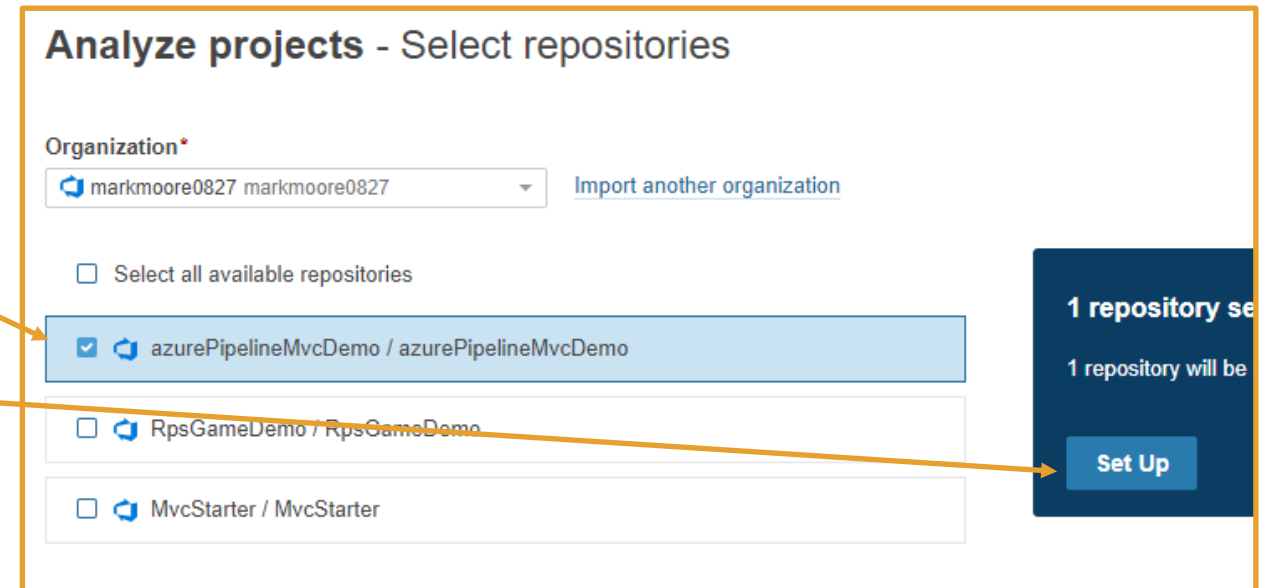
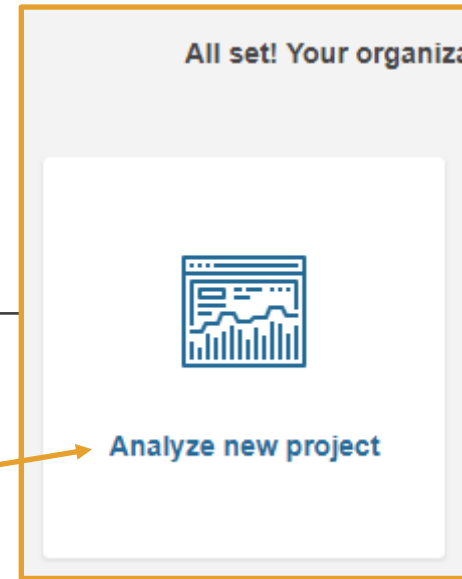
# Analyse a New Project

---

Click 'analyse new project'

Choose the pipeline you want to analyse.

Click 'Set Up'



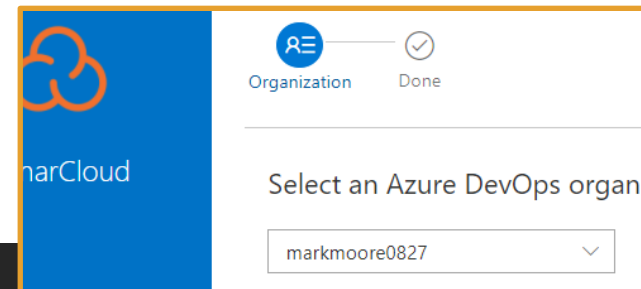
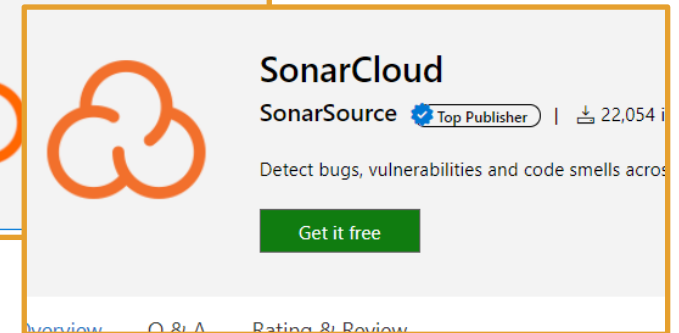
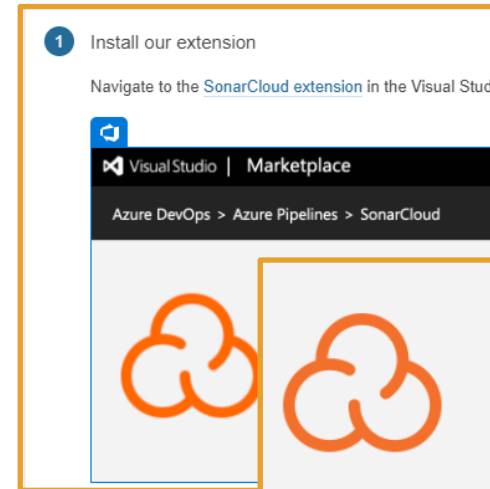
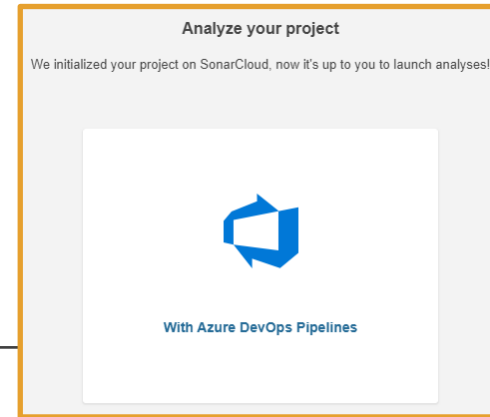


# Configure Code Analysis

Click on 'With Azure DevOps Pipelines'

Click to download the SonarCloud Extension from the Visual Studio Marketplace.

Make sure to choose the correct organization when installing in the Marketplace.



# Configure Azure Pipeline

## 2 Configure Azure Pipeline

What option best describes your build?

Maven

Gradle

.NET

Other (for JS, TS, Go, Python, PHP, ...)



Create a new pipeline

Follow the steps on Azure to initialize your pipeline and link it to your repository.

Add a new SonarCloud Service Endpoint

1 Go to [Project settings > Service connections](#)

2 Add a new service connection of the type SonarCloud

3 Use this token: `f3cf46527df3aec0546d71961a7358d9ecfcfb29`  

4 Click on **Verify** to check that everything is linked correctly.

Add the following three tasks to your pipeline

### 1. Prepare Analysis Configuration

1 Select the SonarCloud endpoint.

2 Select the SonarCloud organization `markmoore0827` 

3 In Choose the way to run the analysis, select **Integrate with MSBuild**.

4 In the Project Key field, enter `markmoore0827_azurePipelineMvcDemo` 

5 In the Project Name field, enter `azurePipelineMvcDemo` 



Please ensure this task runs before your build step.

### 2. Run Code Analysis



This task needs to run after your build step.

These steps will guide you through setup on the Azure DevOps side. See the Step-By-Step on the following slides.

### 2. Run Code Analysis



This task needs to run after your build step.

### 3. Publish Quality Gate Result

This task is not mandatory but will allow you to decorate your Pull Request.

If you plan not to use such a feature, you can omit it. Be aware that this task may increase your build time.

Enable the pipeline on push

Enable [Continuous Integration for your pipeline](#) to make sure your code will be analysed after every update.

Enable the PR Decoration

Enable [PR trigger](#) to make sure your code will be analysed after every update.

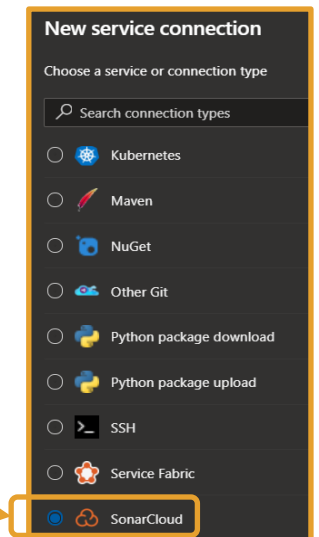
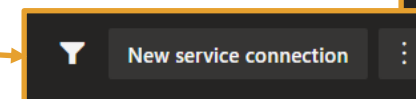
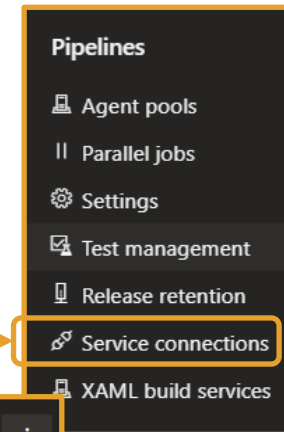
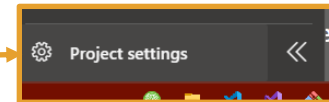


# Add a new SonarCloud Service Endpoint

1. In your Azure DevOps account, click on the project you will be adding the Code Analysis to.
2. Click Project settings at the bottom left of your project home page.
3. Click the 'back' arrow next to Project details.
4. Click Service Connections under Pipelines.
5. Click 'New service connection' in the upper right.
6. Select 'SonarCloud' in the 'New Service Connection' List
7. Click Next.

## Add a new SonarCloud Service Endpoint

- 1 Go to **Project settings** > **Service connections**
- 2 Add a new service connection of the type SonarCloud
- 3 Use this token: `f3cf46527df3aec0546d71961a7358d9ecfcfb29`
- 4 Click on **Verify** to check that everything is linked correctly.



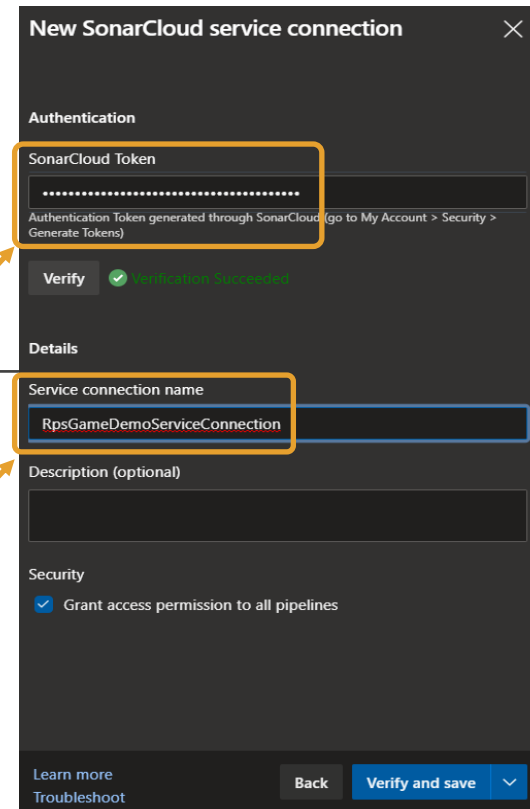
# New SonarCloud Connection

Paste in the P.A.T under 'SonarCloud Token'.

Create a name for your connection under 'Service connection name'

Click 'Verify and Save'.

Verify the connection was successful.



The screenshot shows the 'New SonarCloud service connection' form. It has a dark theme. The 'Authentication' section contains a 'SonarCloud Token' field with a masked value and a 'Verify' button. The 'Details' section contains a 'Service connection name' field with the value 'RpsGameDemoServiceConnection' and a 'Description (optional)' field. The 'Security' section has a checkbox 'Grant access permission to all pipelines' which is checked. At the bottom, there are links for 'Learn more' and 'Troubleshoot', and buttons for 'Back' and 'Verify and save'.

**New SonarCloud service connection**

**Authentication**

SonarCloud Token  
.....  
Authentication Token generated through SonarCloud (go to My Account > Security > Generate Tokens)

Verify ✓ Verification Successful

**Details**

Service connection name  
RpsGameDemoServiceConnection

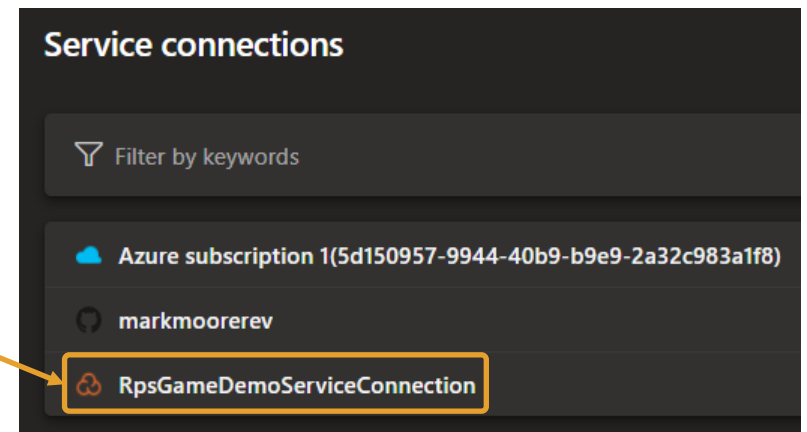
Description (optional)  
.....

**Security**

☒ Grant access permission to all pipelines

Learn more  
Troubleshoot

Back Verify and save



# Add publish task to your pipeline YAML

**1. Prepare Analysis Configuration**

- 1 Select the SonarCloud endpoint.
- 2 Select the SonarCloud organization `markmoore0827`
- 3 In Choose the way to run the analysis, select **Integrate with MSBuild**.
- 4 In the Project Key field, enter `markmoore0827_azurePipelineMvcDemo`
- 5 In the Project Name field, enter `azurePipelineMvcDemo`

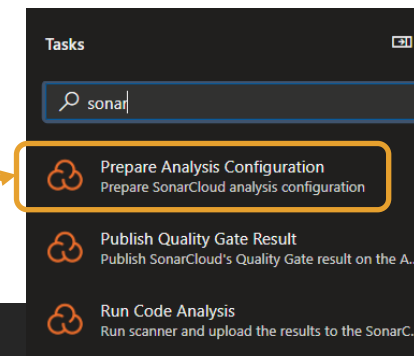
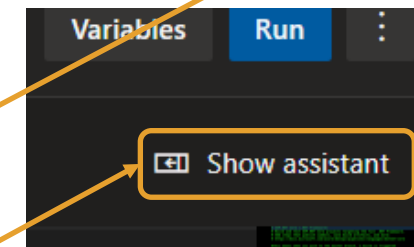
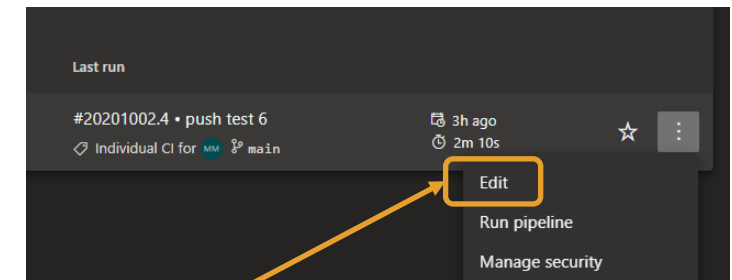
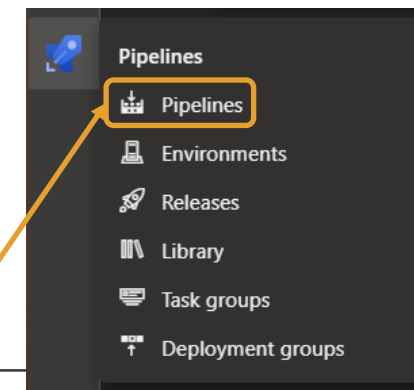
*Please ensure this task runs before your build step.*

Click Pipelines under the Rocket (Pipelines) avatar in the left column.

Click Edit under the pipeline you want to add code analysis to.

Click 'Show Assistant' in the upper left.

Select 'Prepare Analysis Configuration'



# Add Code Analysis configuration task to your pipeline YAML

## 1. Prepare Analysis Configuration

- 1 Select the SonarCloud endpoint.
- 2 Select the SonarCloud organization `markmoore0827`
- 3 In Choose the way to run the analysis, select **Integrate with MSBuild**.
- 4 In the Project Key field, enter `markmoore0827_azurePipelineMvcDemo`
- 5 In the Project Name field, enter `azurePipelineMvcDemo`



Please ensure this task runs before your build step.

You don't have to change anything under 'Advanced'.  
Place the cursor on the line above your 'build' Task.  
Click 'Add'

Prepare Analysis Configuration

SonarCloud Service Endpoint \* ⓘ  
RpsGameDemoServiceConnection

Organization \* ⓘ  
markmoore0827 (markmoore0827)

Choose the way to run the analysis \* ⓘ  
☒ Integrate with MSBuild  
☐ Integrate with Maven or Gradle  
☐ Use standalone scanner

Project Key \* ⓘ  
markmoore0827\_azurePipelineMvcDemo

Project Name ⓘ  
azurePipelineMvcDemo

Project Version ⓘ  
1.0

```
Settings
- task: SonarCloudPrepare@1
  inputs:
    SonarCloud: 'RpsGameDemoServiceConnection'
    organization: 'markmoore0827'
    scannerMode: 'MSBuild'
    projectKey: 'markmoore0827_azurePipelineMvcDemo'
    projectName: 'azurePipelineMvcDemo'

Settings
- task: DotNetCoreCLI@2
  displayName: Build
  inputs:
```

# Add Code Analysis to your YAML

## 2. Run Code Analysis



This task needs to run after your build step.

Click 'Show Assistant' to return to the Assistant.

Make sure the cursor is immediately below the 'build' task.

Click 'Run Code Analysis'.

### Tasks



sonar



Prepare Analysis Configuration

Prepare SonarCloud analysis configuration



Publish Quality Gate Result

Publish SonarCloud's Quality Gate result on the A...



Run Code Analysis

Run scanner and upload the results to the SonarC...

### Settings

```
- task: DotNetCoreCLI@2
  displayName: Build
  inputs:
    command: 'build'
    projects: './RpsGame/RPS_GameMv
    arguments: '--configuration $(b
```

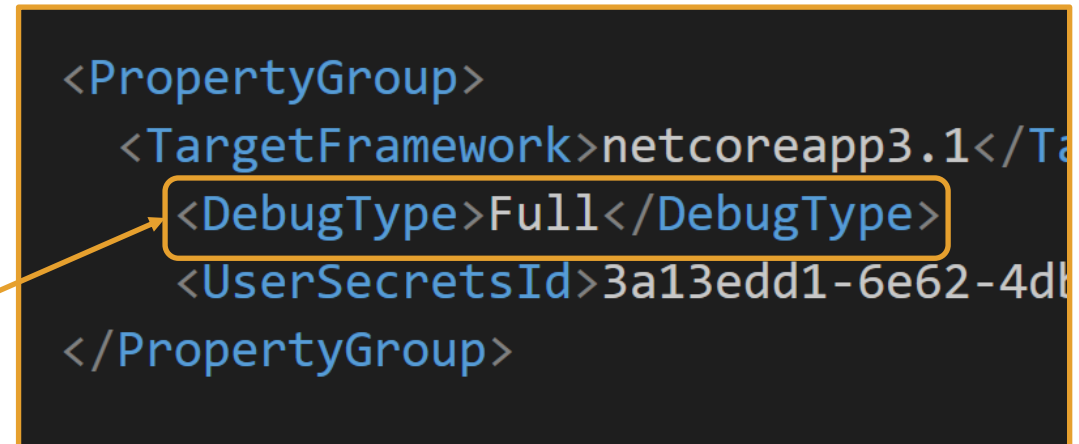
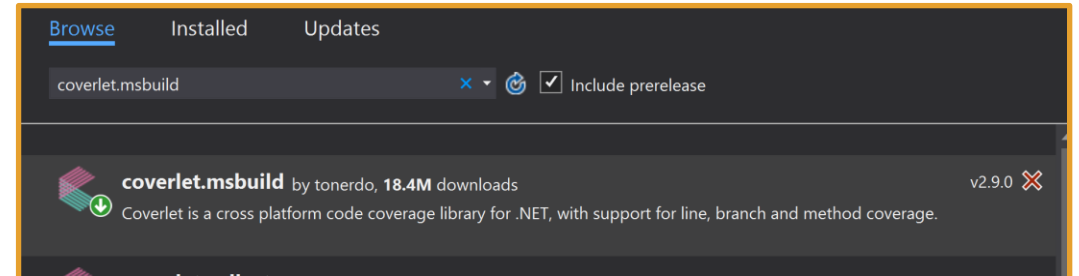
### Settings

```
- task: SonarCloudAnalyze@1
```

# Add Coverlet and DebugType to your project

For the code coverage analysis and report to work, your test project needs to use the '**coverlet.msbuild**' package. (it may be possible to use something else, but this is what I used)

1. Install '**coverlet.msbuild**' to ONLY your test project in Visual Studio.
2. For each project, click the project name. Under **<PropertyGroup>**, add **<DebugType>Full</DebugType>** for the **coverlet.msbuild** tool to function properly.

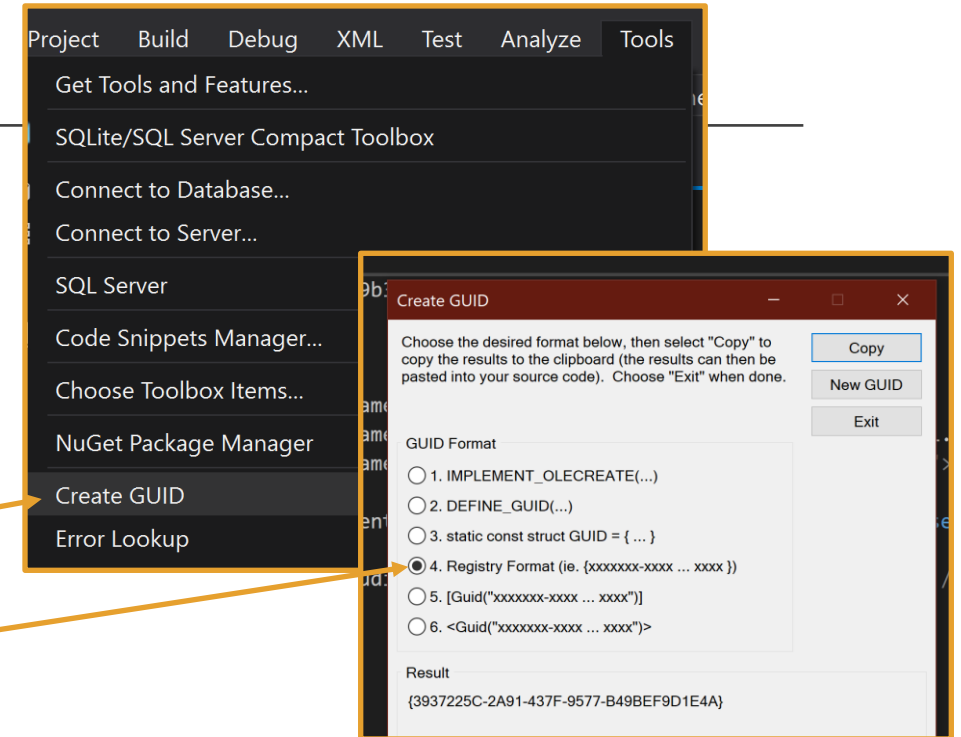


# Add the Project GUID Property to your projects

Each Project needs a

`<ProjectGuid>#</ProjectGuid>` property so that SonarCloud can differentiate between them and analyze them properly.

1. To generate a GUID, click on **Tools** in the menu bar in Visual Studio.
2. Click '**Create GUID**'.
3. Select '**4. Registry Format**'.
4. Copy the GUID
5. Add the GUID to a new
  1. `<ProjectGuid>` property in the Project.



```
<PropertyGroup>
  <TargetFramework>netcoreapp3.1</TargetFramework>
  <DebugType>Full</DebugType>
  <UserSecretsId>3a13edd1-6e62-4db1-9578-572a9b314abd</UserSecretsId>
  <ProjectGuid>{EDC476A1-A5C3-4C7D-BC18-469A28F48169}</ProjectGuid>
</PropertyGroup>
```



# Darius Vallejo example

---

Darius Vallejo based his integration on the below tutorial.

<https://azuredevopslabs.com/labs/vstsextend/sonarcloud//>