SonarQube

SonarQube an open source platform for continuous inspection of code quality to perform automatic reviews with static analysis of code to:

- Detect Bugs
- Code Smells
- Security Vulnerabilities
- Centralize Quality

Setup SonarQube on Azure and integrate with Azure DevOps project

In Azure CLI

Create a Resource Group. Replace `<region>` with the region of your choosing, for example, eastus.

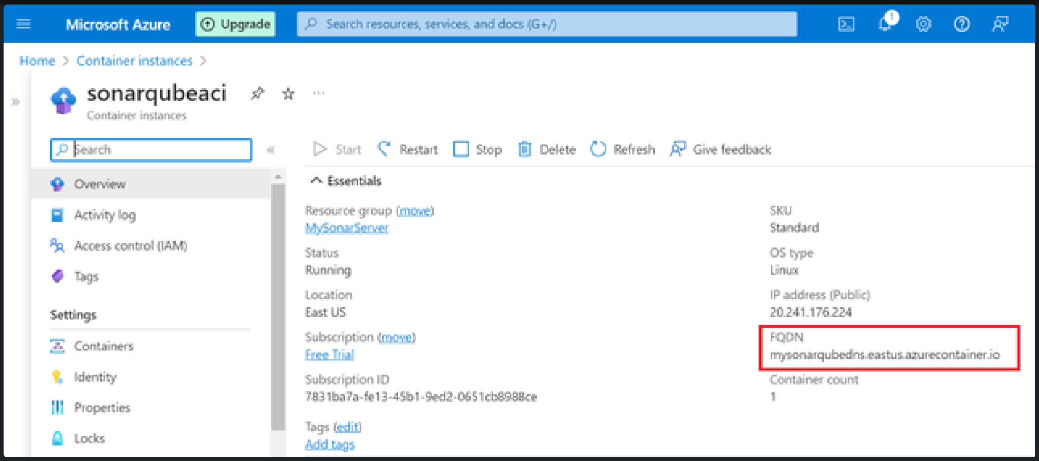
```
az group create --name MySonarServer --location eastus
```

Create Azure Container Instance with official SonarQube Docker image:

```
az container create -g MySonarServer `
--name sonarqubeaci `
--image sonarqube `
--ports 9000 `
--dns-name-label mysonarqubedns `
--cpu 2 `
--memory 3.5
```

Name	Description
`--name`	Name of the container instance.
`--image`	The container image name. Here we are fetching official SonarQube image from DockerHub
`--dns-name-label`	The dns name label for container with public IP.
`--ports`	The ports to open. The default port for SoanrQube is `9000`. We need to expose this port to access SonarQube.
`--cpu`	The required number of CPU cores of the containers.
`--memory`	The required memory of the containers in GB

In Azure Portal



Resource group (move)
MySonarServer

Status
Running

Location
East US

Subscription (move)
Free Trial

Subscription ID
7831ba7a-fe13-45b1-9ed2-0651cb8988ce

Tags (edit)
Add tags

SKU
Standard

OS type
Linux

IP address (Public)
20.241.176.224

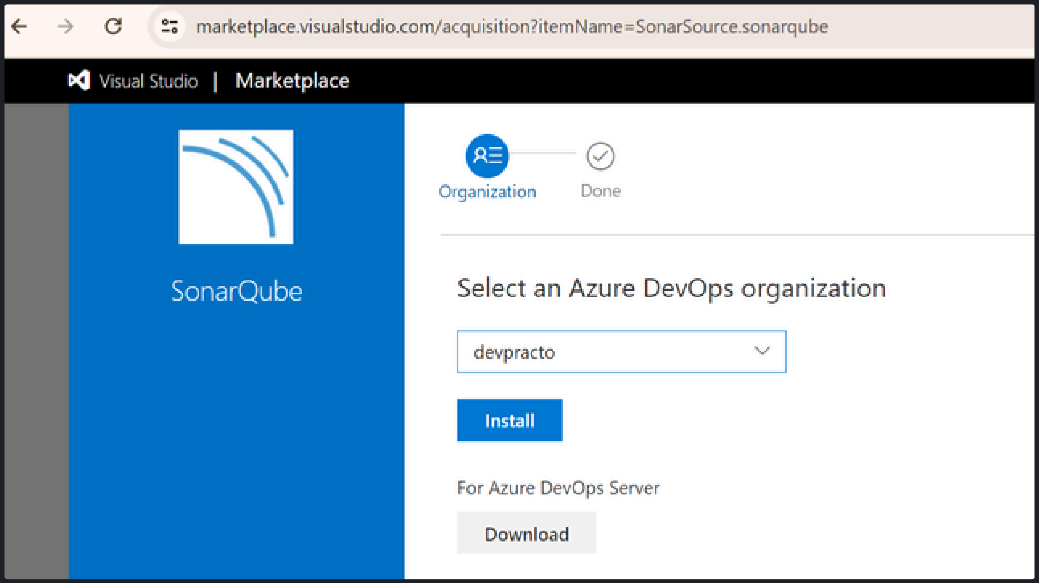
FQDN
mysonarqubedns.eastus.azurecontainer.io

Container count
1

In DevOps Portal

Install sonarQube extension

https://marketplace.visualstudio.com/items?itemName=SonarSource.sonarqube


marketplace.visualstudio.com/acquisition?itemName=SonarSource.sonarqube
Visual Studio | Marketplace
SonarQube
Organization
Done
Install
Download

Select an Azure DevOps organization

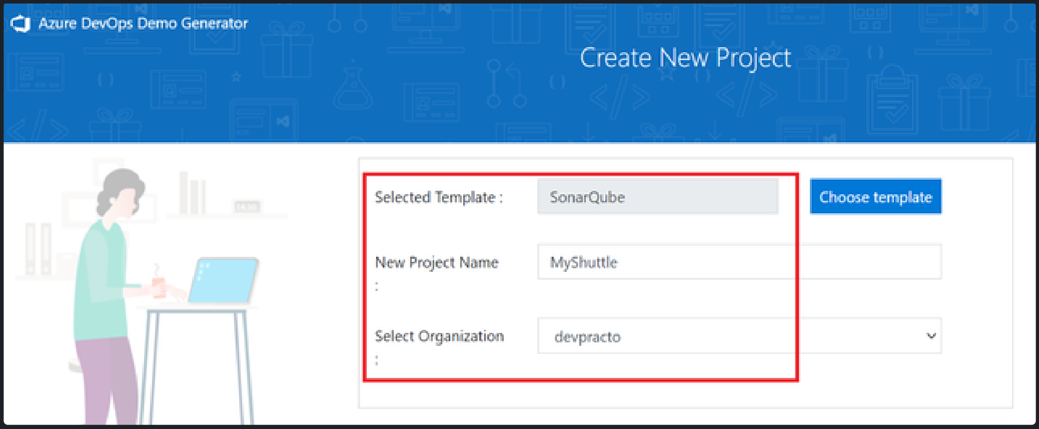
devpracto

For Azure DevOps Server

In Azure DevOps Portal

Use the Azure DevOps Demo Generator to provision a project on your Azure DevOps Organization:

https://azuredevopsdemogenerator.azurewebsites.net/?TemplateId=77364&Name=SonarQube



Selected Template : SonarQube

New Project Name : MyShuttle

Select Organization : devpracto

In any browser

Open a browser and login to the SonarQube Portal using the following credentials:

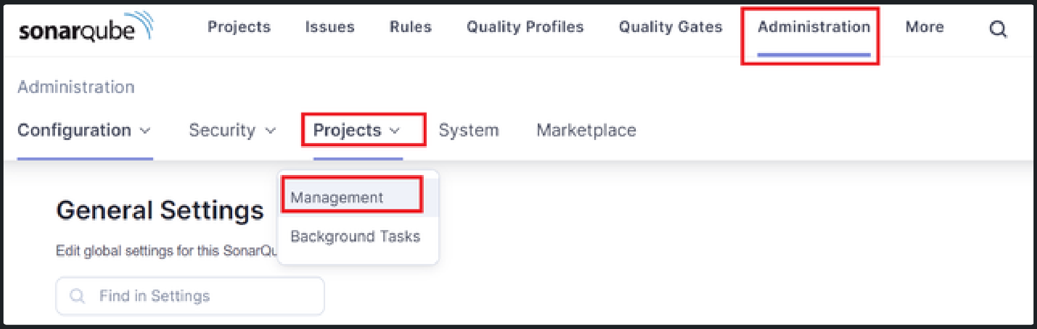
http://mysonarqubedns.eastus.azurecontainer.io:9000/

Username= admin, Password= admin; Change the password to a different value

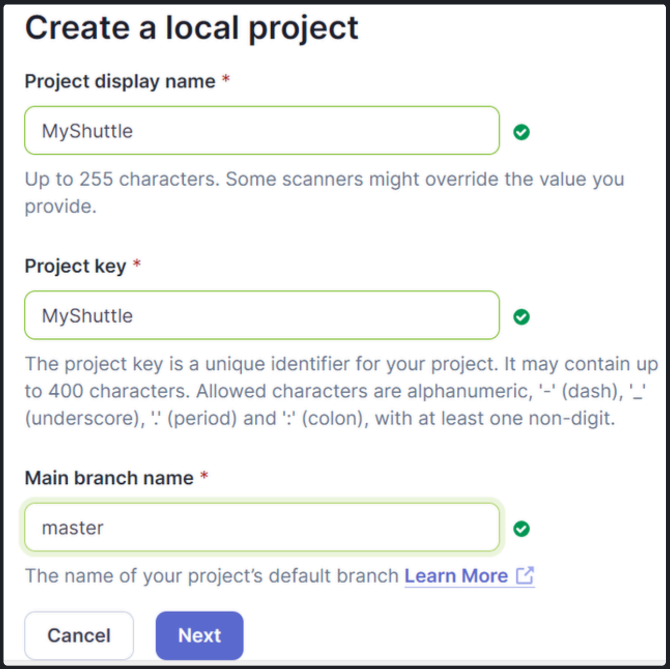
You are now in SonarQube Portal!

Create a SonarQube Project and configure Quality Gate

Choose **Administration** in the toolbar, click **Projects** tab and then **Management**.



Create a project with **Name** and **Key** as **MyShuttle**. Provide **Main branch name** as **master**. Then click on **Create.**


Create a local project

Project display name *

MyShuttle

Up to 255 characters. Some scanners might override the value you provide.

Project key *

MyShuttle

The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.

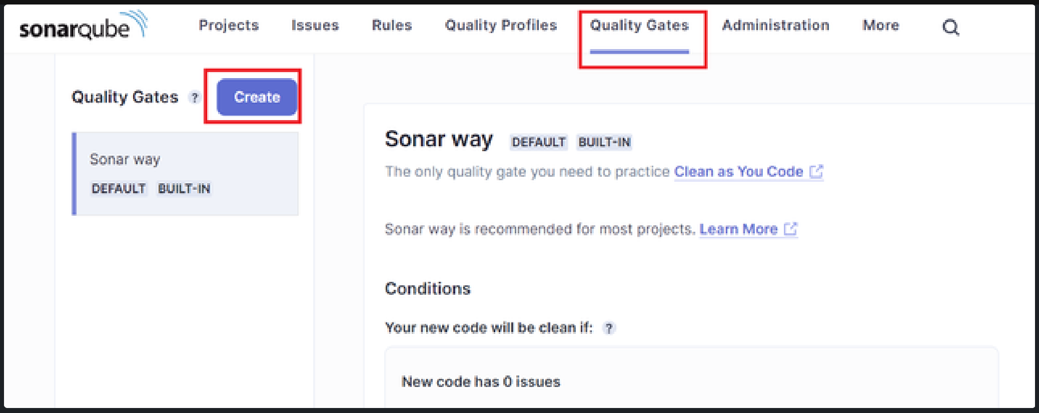
Main branch name *

master

The name of your project's default branch Learn More

Create a Quality Gate to enforce a policy which fails the gate if there are bugs in the code.

A Quality Gate is a PASS/FAIL check on a code quality that must be enforced before releasing software.



Sonar way DEFAULT BUILT-IN

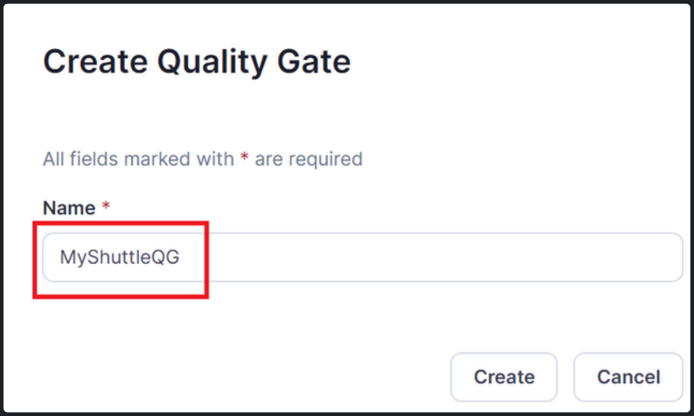
The only quality gate you need to practice Clean as You Code

Sonar way is recommended for most projects. Learn More

Conditions

Your new code will be clean if:

New code has 0 issues



Create Quality Gate

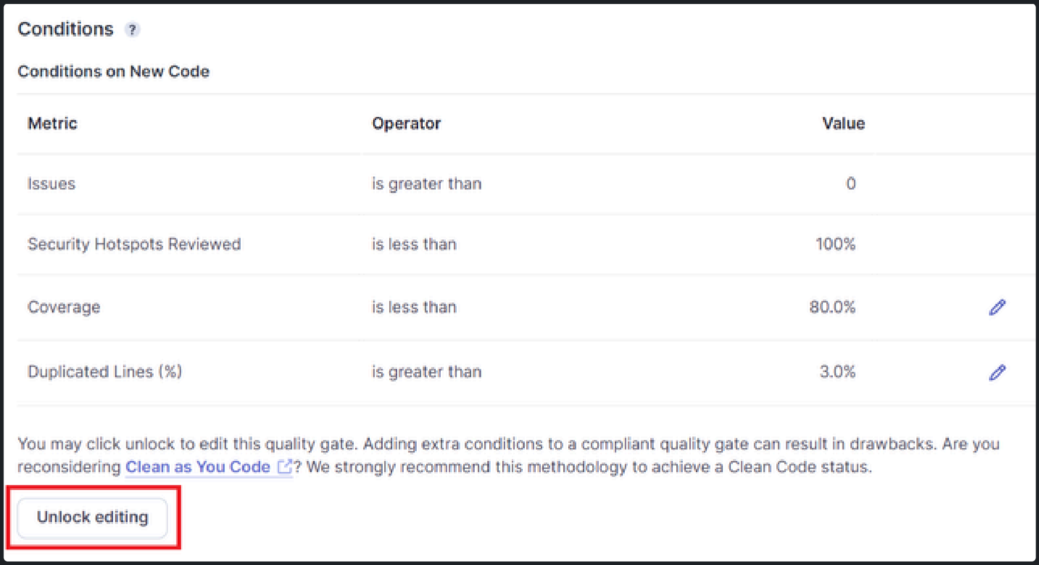
All fields marked with * are required

Name *

MyShuttleQG

Add a condition to check for the number of bugs in the code.

Click Unlock editing



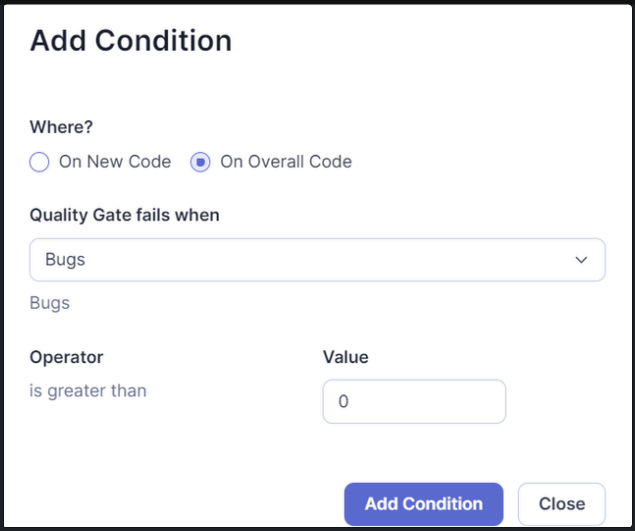
Conditions

Conditions on New Code

Metric	Operator	Value
Issues	is greater than	0
Security Hotspots Reviewed	is less than	100%
Coverage	is less than	80.0%
Duplicated Lines (%)	is greater than	3.0%

You may click unlock to edit this quality gate. Adding extra conditions to a compliant quality gate can result in drawbacks. Are you reconsidering Clean as You Code? We strongly recommend this methodology to achieve a Clean Code status.

Click on **Add Condition** as shown


Add Condition

Where?

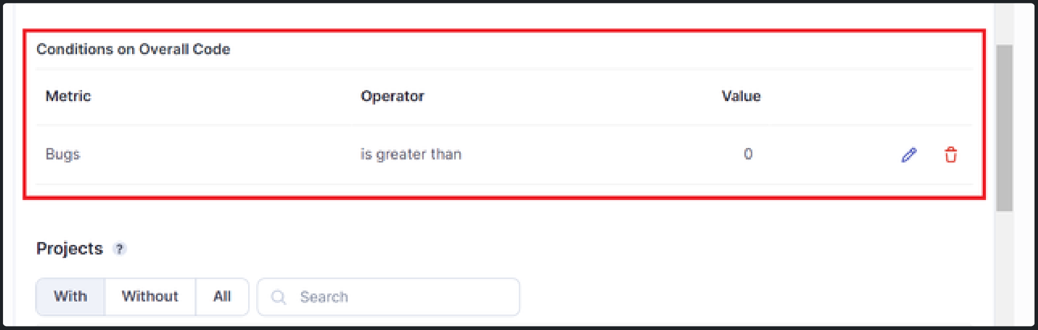
On New Code [✓ On Overall Code]

Quality Gate fails when

Bugs

Bugs

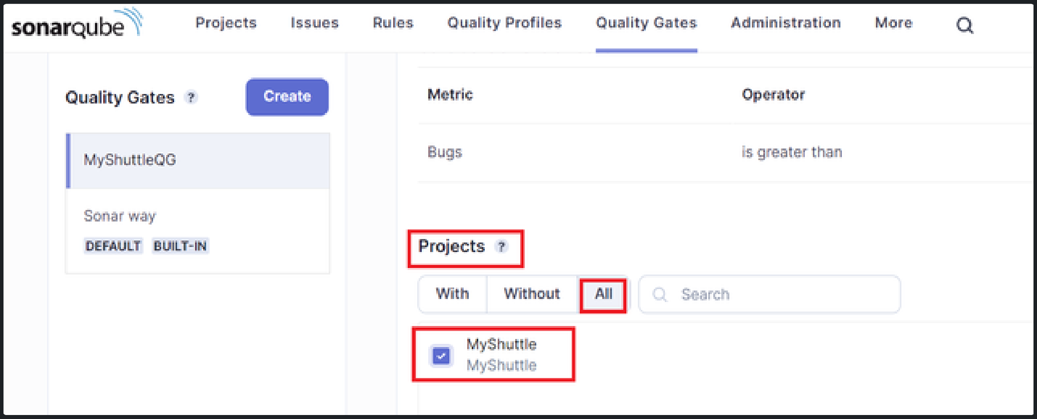
Operator	Value
is greater than	0


Conditions on Overall Code

Metric	Operator	Value
Bugs	is greater than	0

Enforce this quality gate for **MyShuttle** project

Click on **All** under **Projects** section and select the project checkbox.

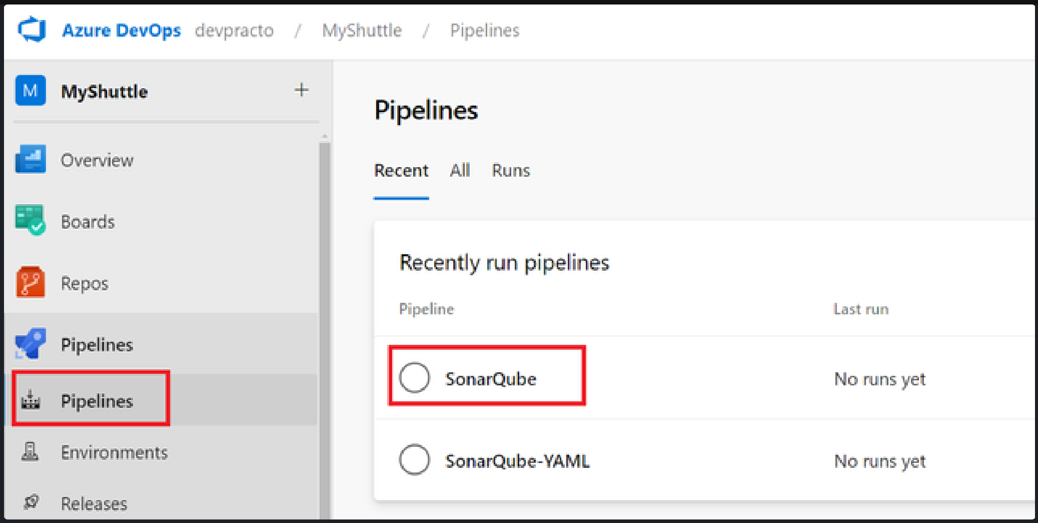


Metric	Operator
Bugs	is greater than

[✓] MyShuttle MyShuttle

Modify the Build to Integrate with SonarQube

Modify Azure Build pipeline to integrate with SonarQube to analyze the java code provisioned by the Azure DevOps Demo Generator system. This is a Java application and we are using Maven to build the code. We are using SonarQube extension tasks to prepare analysis on SonarQube and publish Quality Gate results.

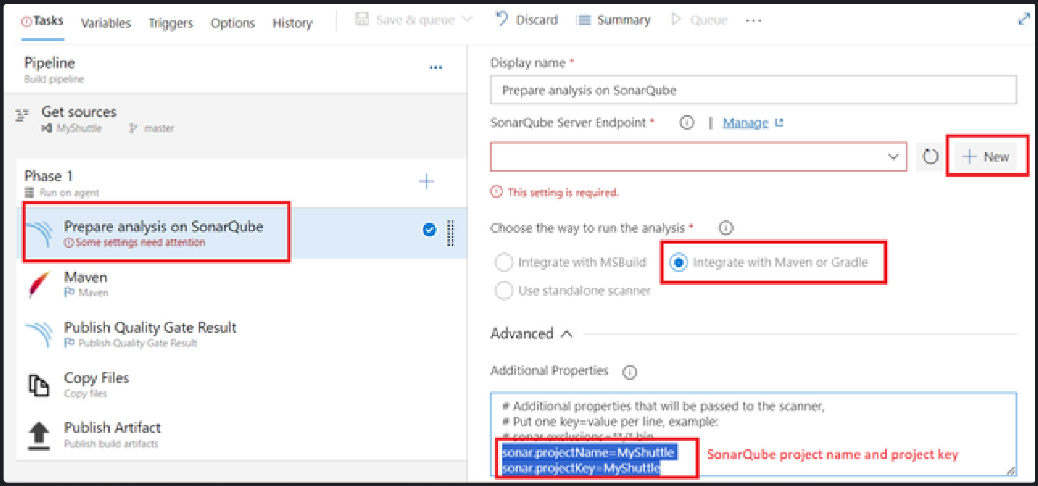


Recently run pipelines

Pipeline	Last run
SonarQube	No runs yet
SonarQube-YAML	No runs yet

Go to **pipelines** under **Pipelines** tab, edit the build pipeline **SonarQube**.

Prepare Analysis Configuration task is to configure all the required settings before executing the build.



Display name * : Prepare analysis on SonarQube

SonarQube Server Endpoint * :

This setting is required.

Choose the way to run the analysis * : Integrate with MSBuild [✓ Integrate with Maven or Gradle] Use standalone scanner

Advanced

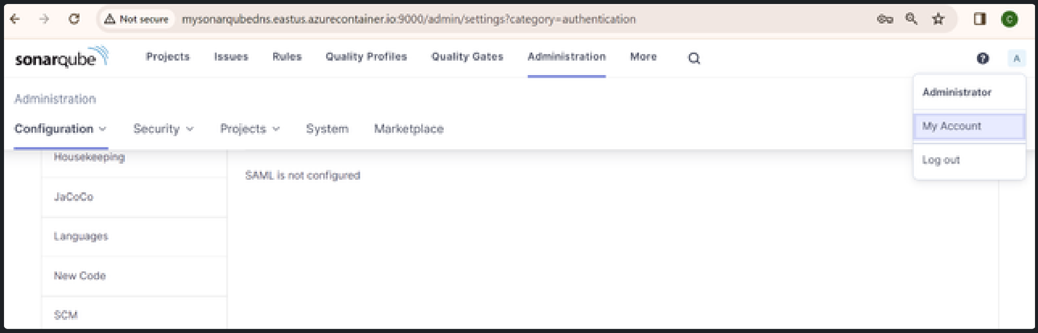
Additional Properties

```
# Additional properties that will be passed to the scanner,
# Put one key=value per line, example:
# sonar.exclusions=**/*.bin
sonar.projectName=MyShuttle
sonar.projectKey=MyShuttle
```

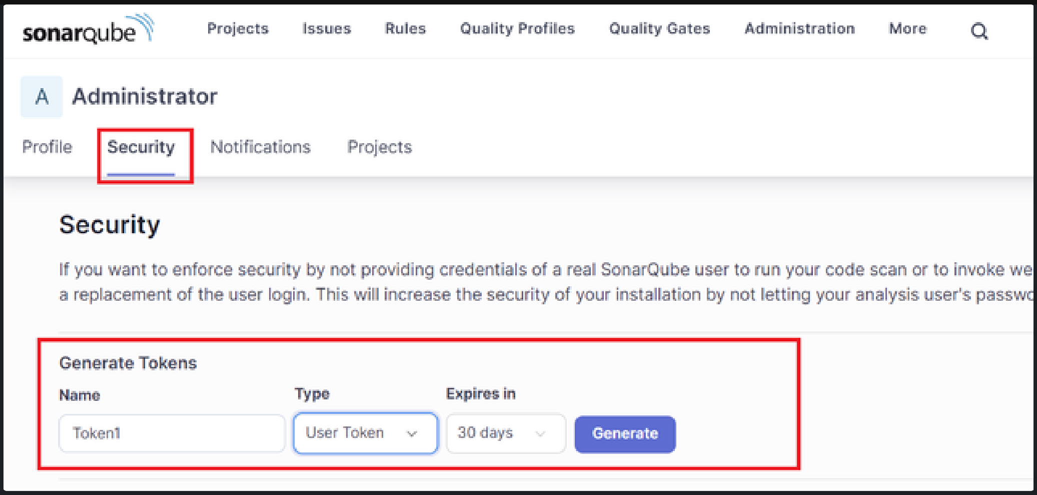
SonarQube project name and project key

In SonarQube Portal

We need a SonarQube Token. Generate as:



SAML is not configured



Security

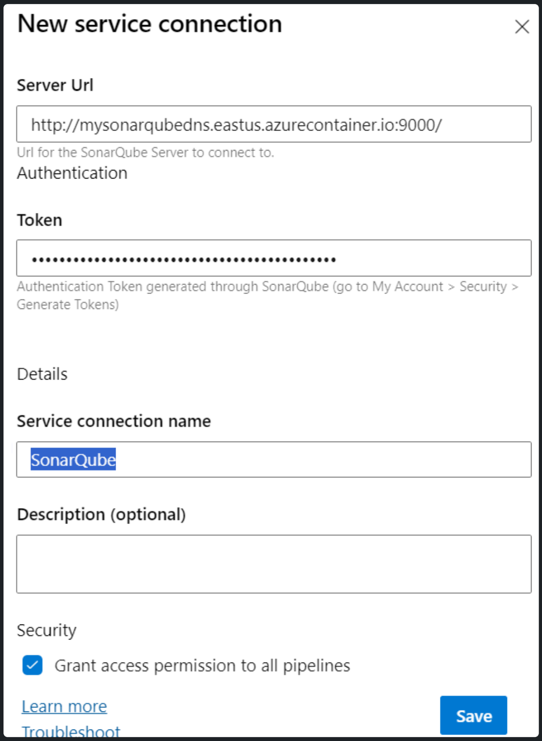
If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke we
a replacement of the user login. This will increase the security of your installation by not letting your analysis user's passwo

Generate Tokens

Name: Token1
Type: User Token
Expires in: 30 days

Click **+ NEW** to add SonarQube server endpoint.

In the **Add SonarQube service connection** wizard enter the SonarQube server URL and SonarQube security token detials.



New service connection

Server Url
http://mysonarqubedns.eastus.azurecontainer.io:9000/
Url for the SonarQube Server to connect to.

Authentication

Token
••
Authentication Token generated through SonarQube (go to My Account > Security > Generate Tokens)

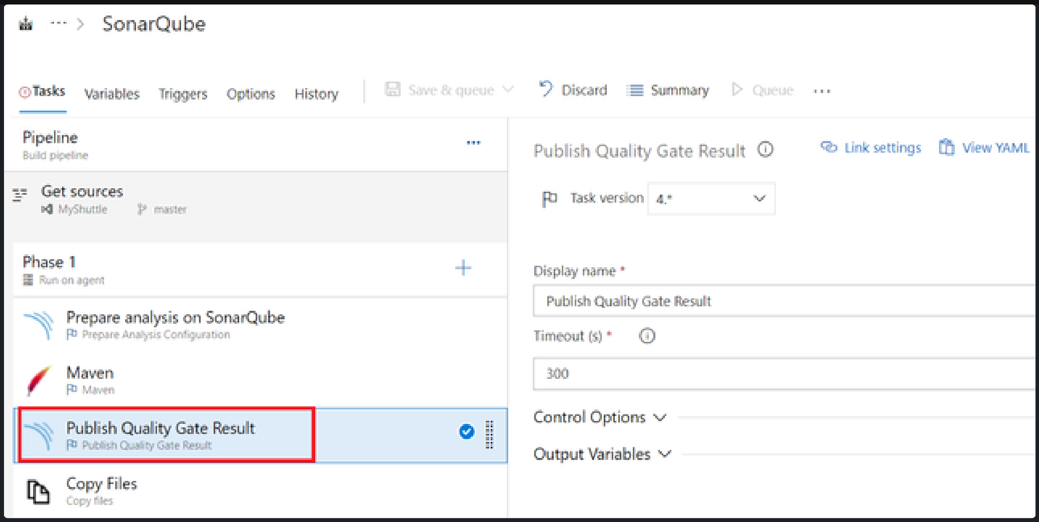
Details

Service connection name
SonarQube

Description (optional)

Security
✓ Grant access permission to all pipelines

Publish Quality Gate Result task is to display the Quality Gate status in the build summary.



Publish Quality Gate Result

Task version 4.*

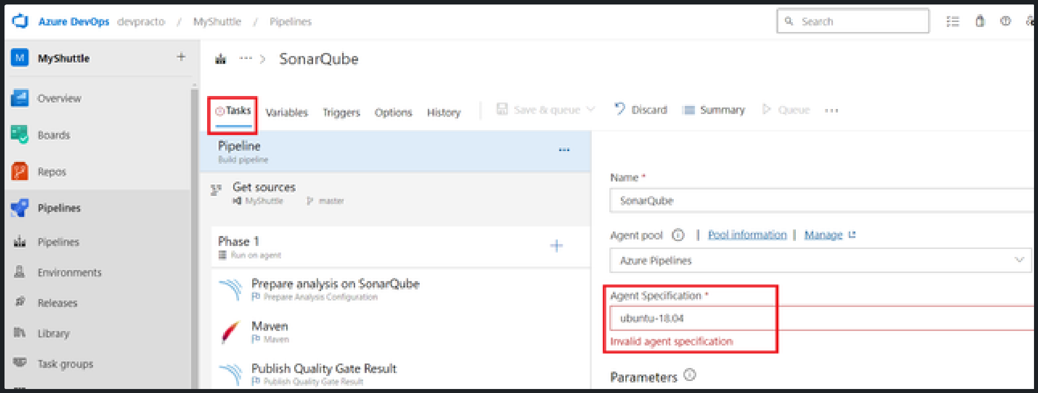
Display name * Publish Quality Gate Result

Timeout (s) * 300

Control Options

Output Variables

Go to Tasks



Name * SonarQube

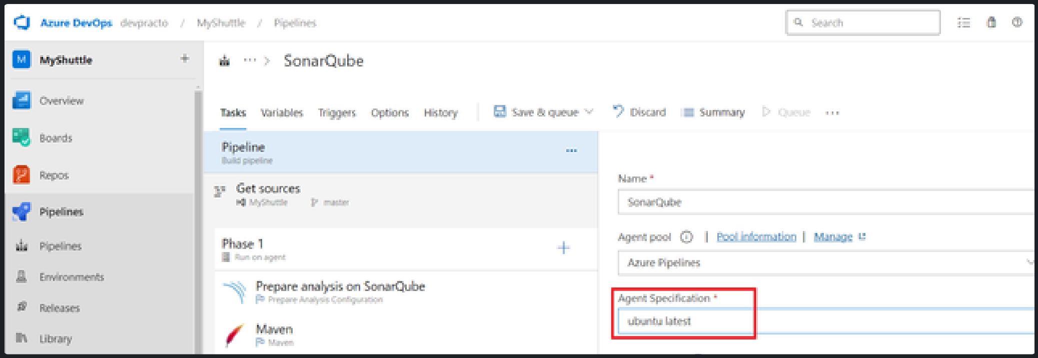
Agent pool Azure Pipelines

Agent Specification * ubuntu-18.04

Invalid agent specification

Parameters

Change to ubuntu:latest

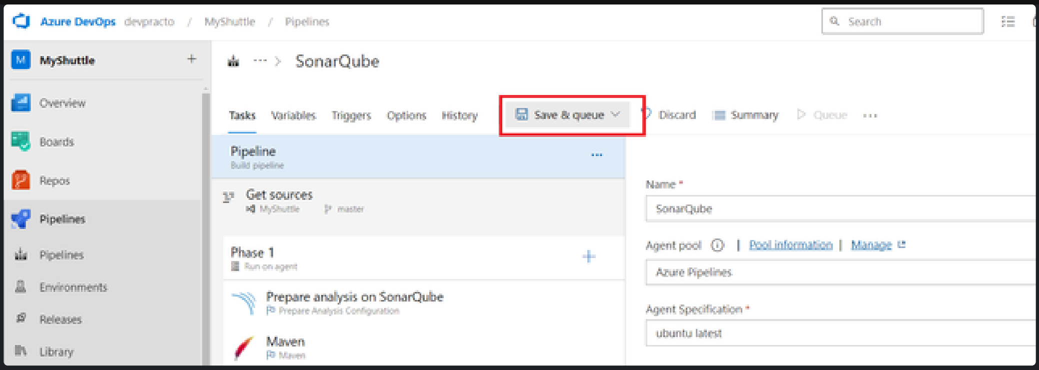


Name * SonarQube

Agent pool Azure Pipelines

Agent Specification * ubuntu latest

Click Save & queue



Name *
SonarQube

Agent pool | Pool information | Manage
Azure Pipelines

Agent Specification *
ubuntu latest

You will see that the build has succeeded but the associated **SonarQube Quality Gate** has **failed**.

The count of bugs is also displayed under **SonarQube Analysis Report**.