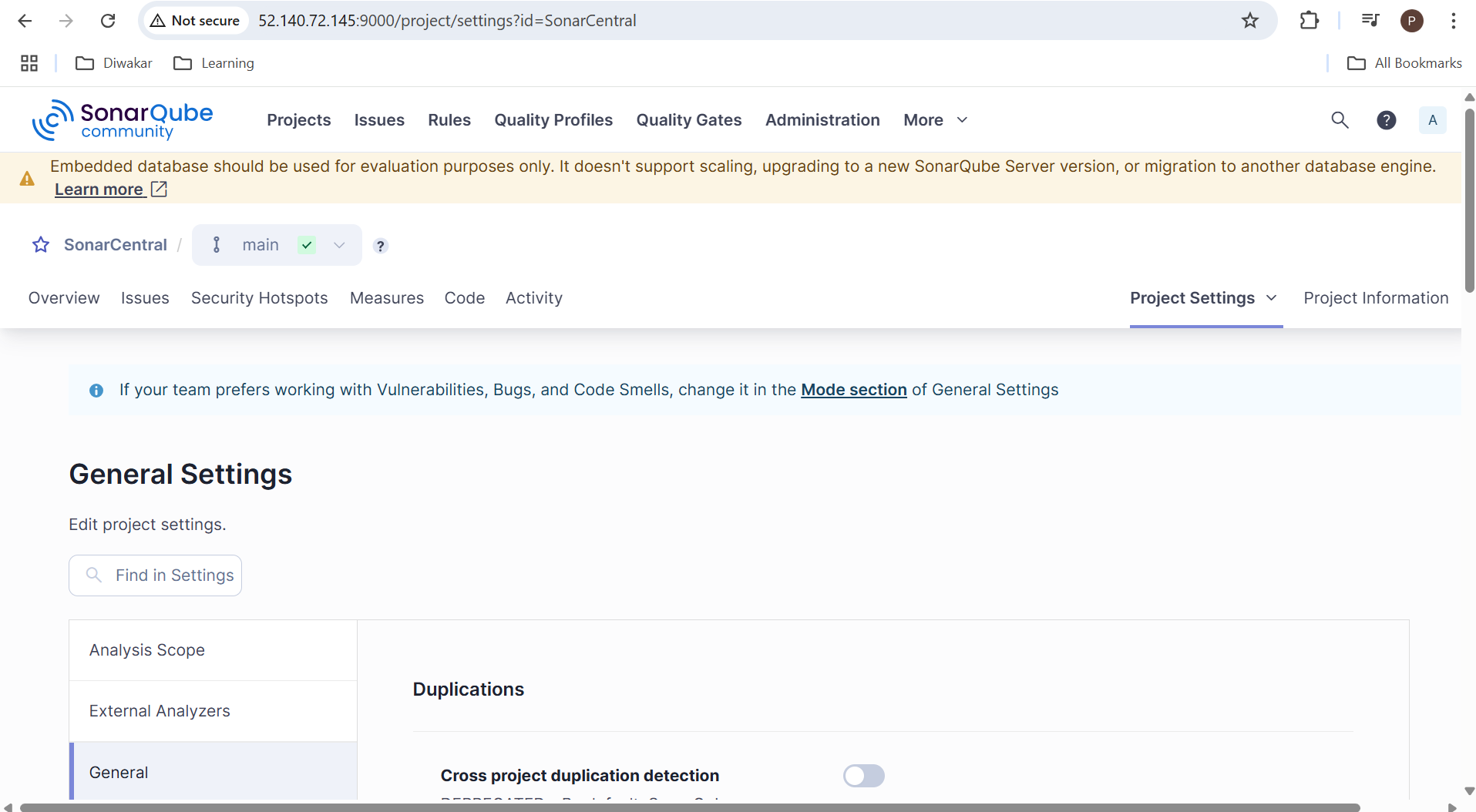
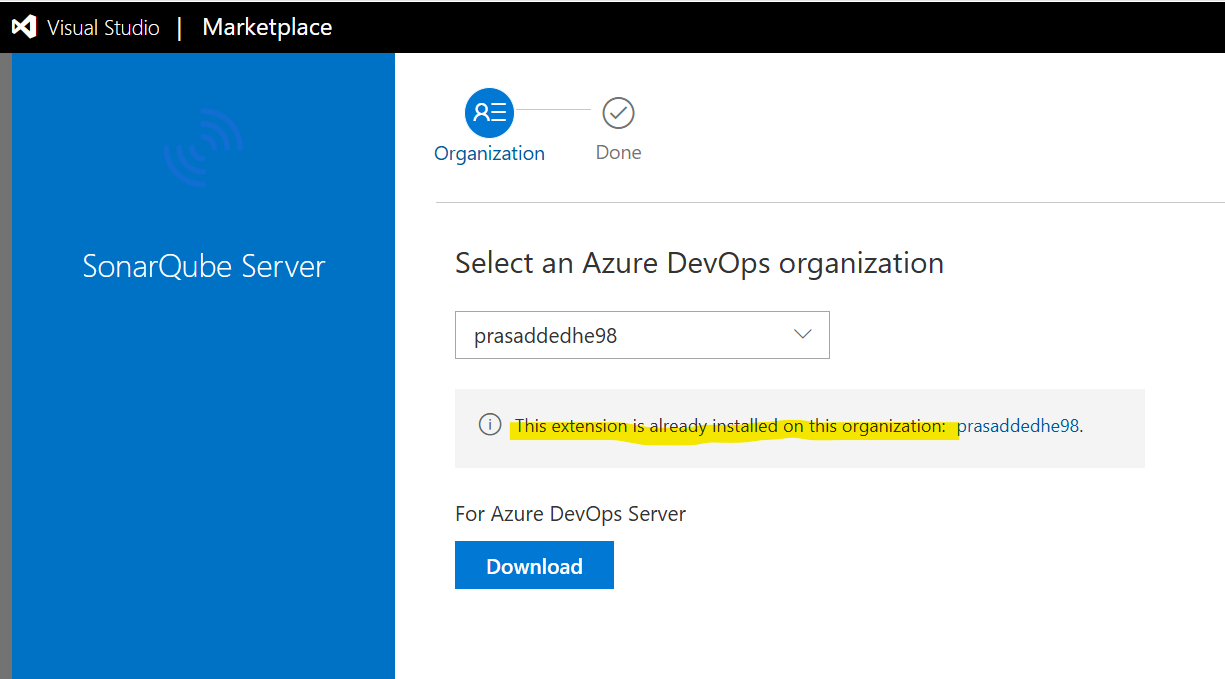
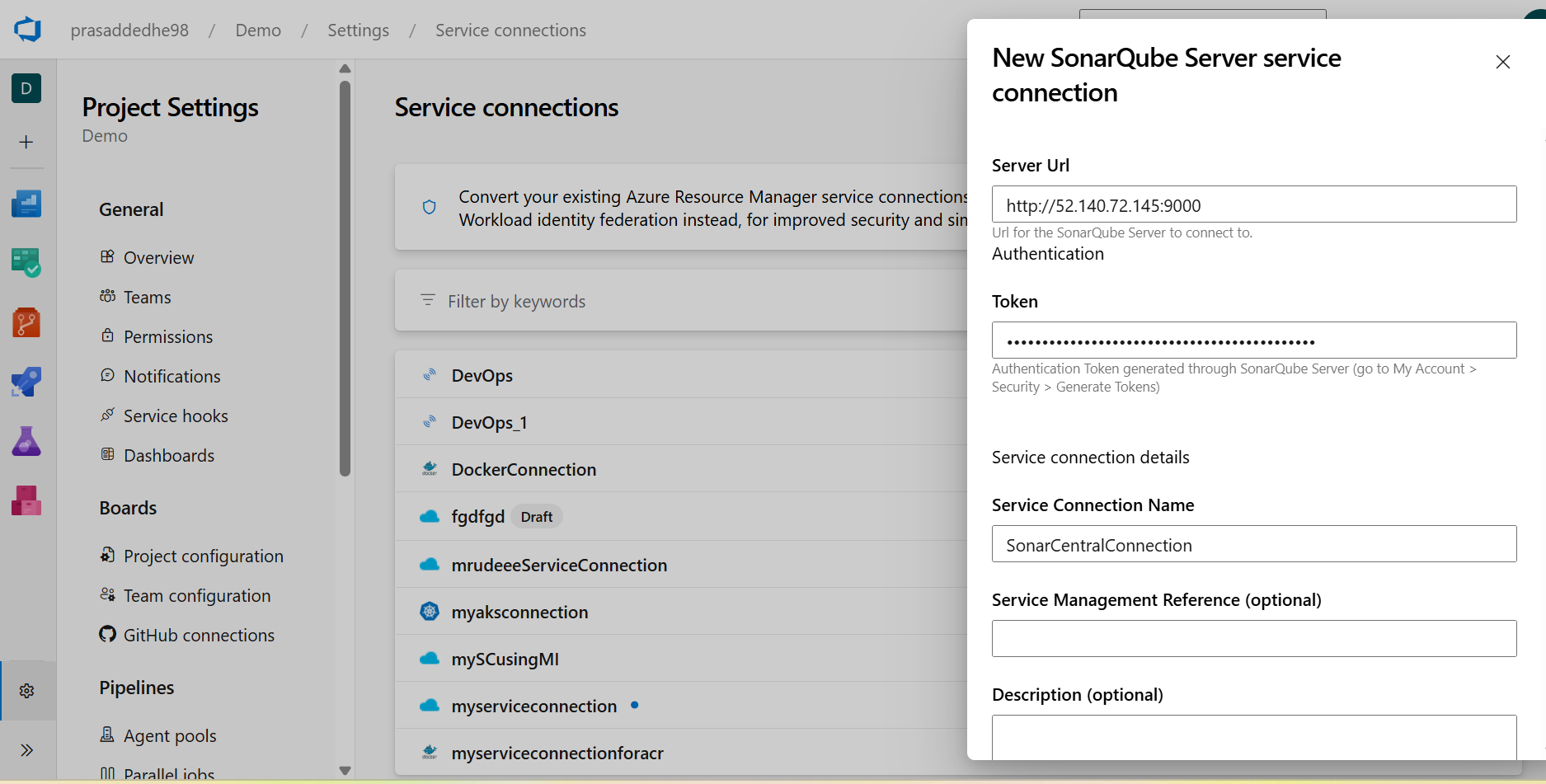
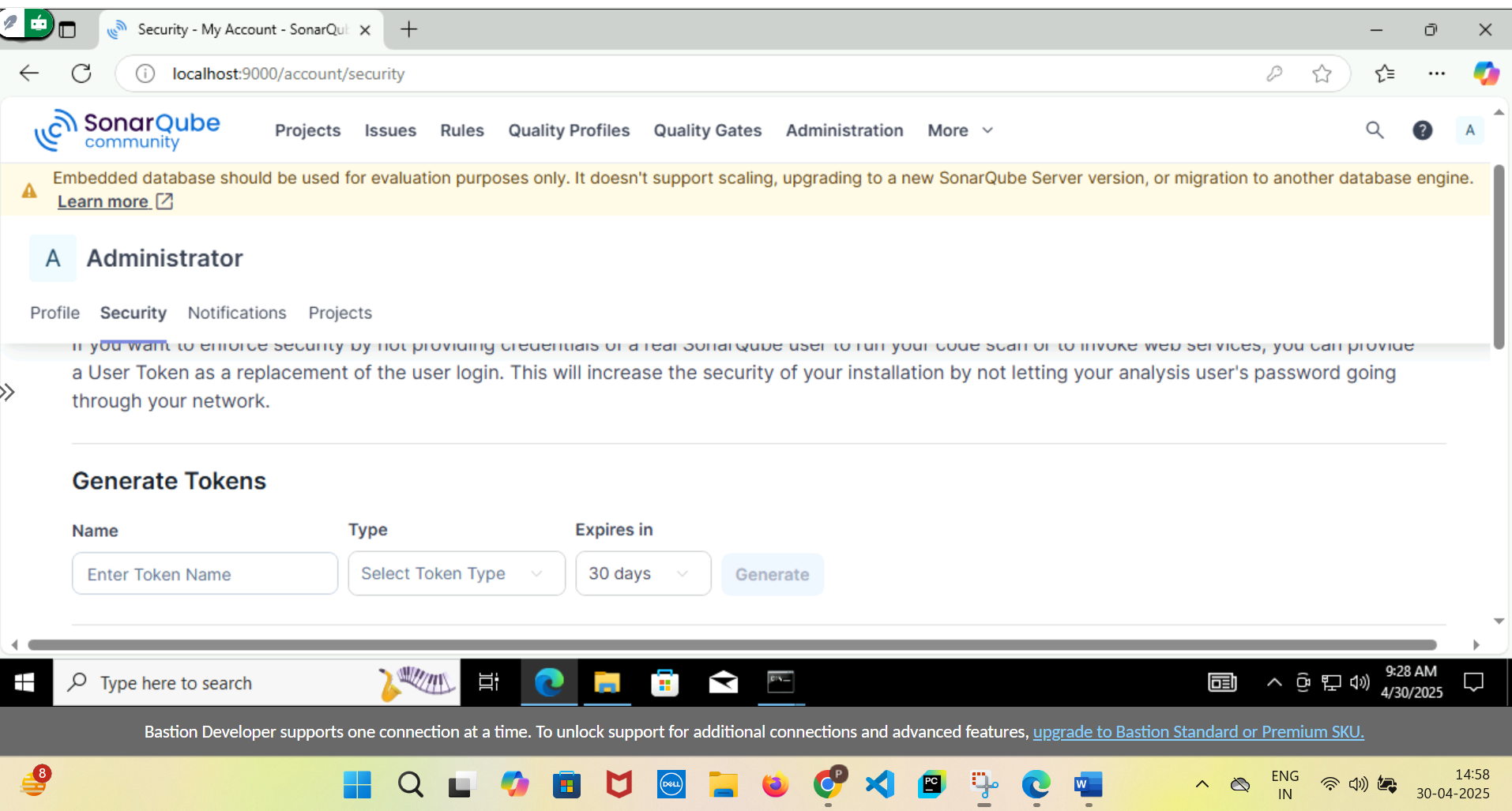
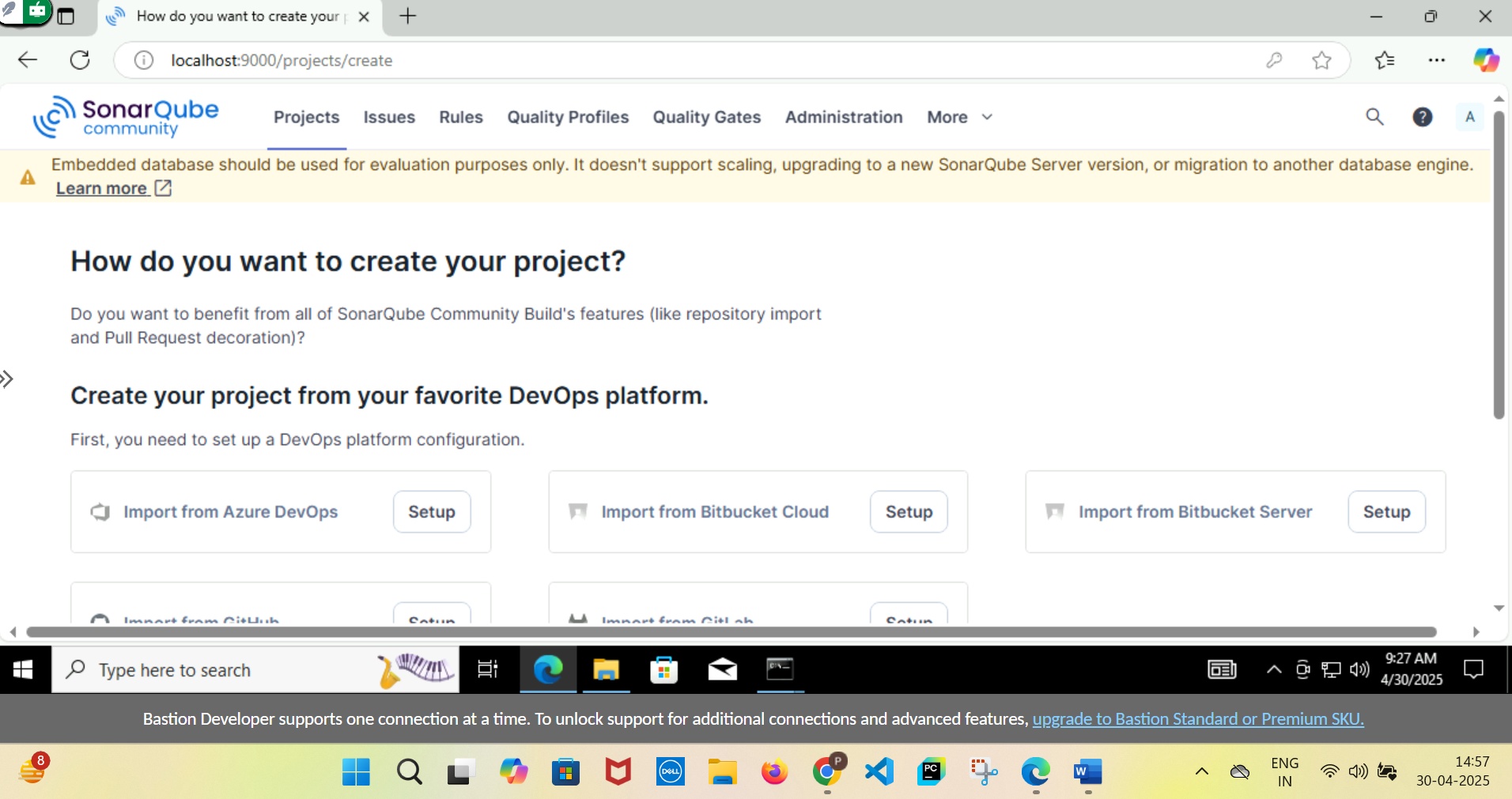
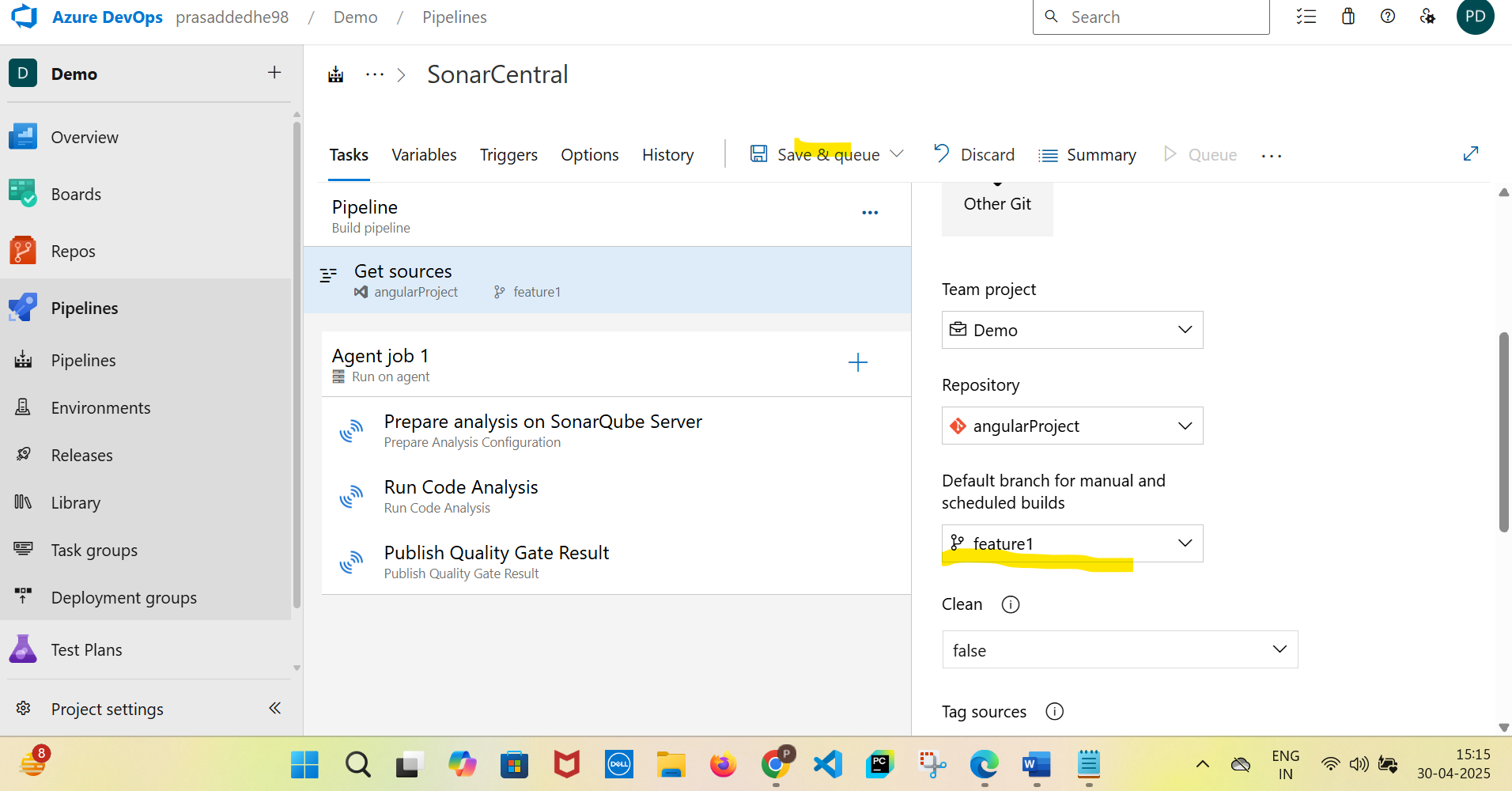
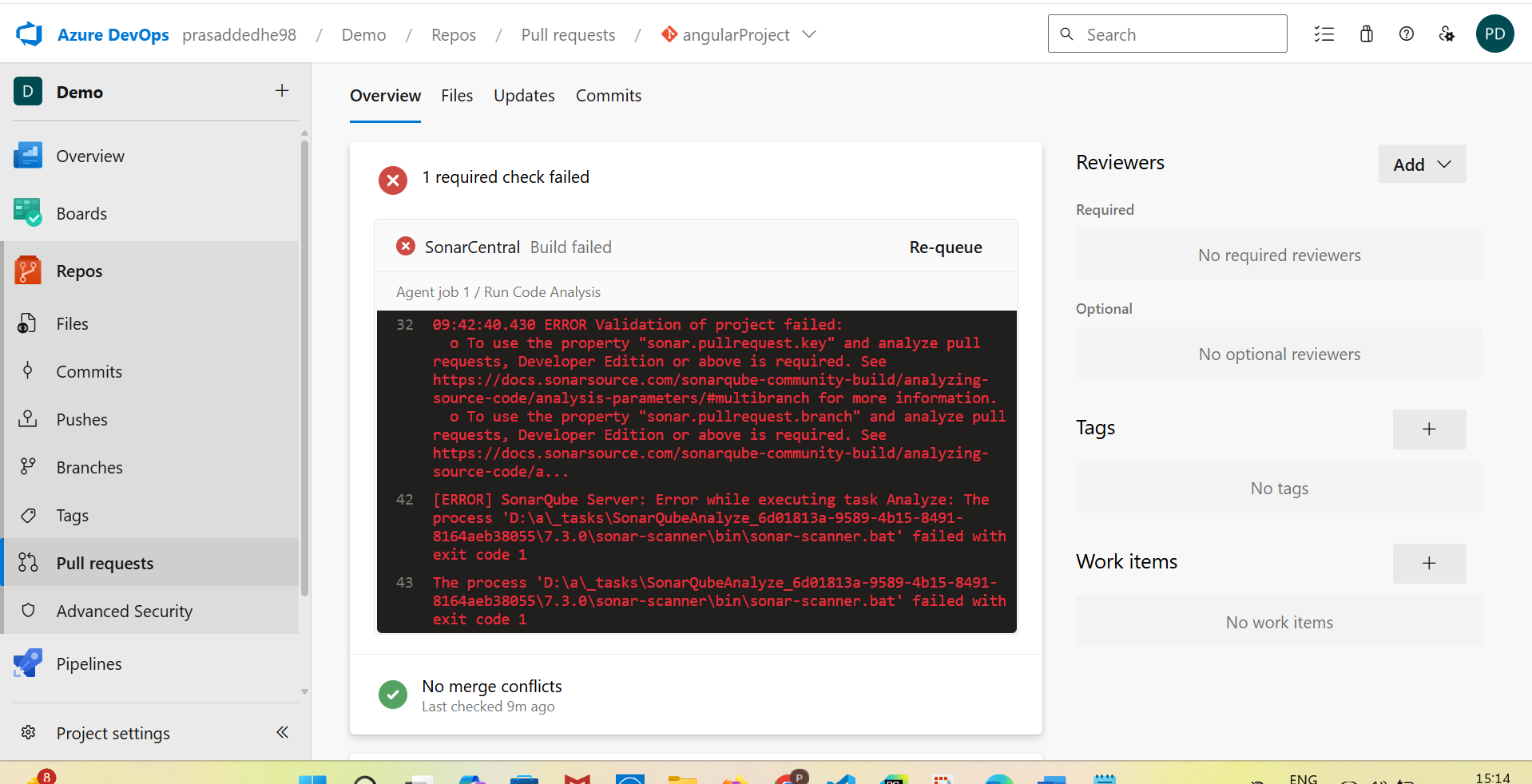
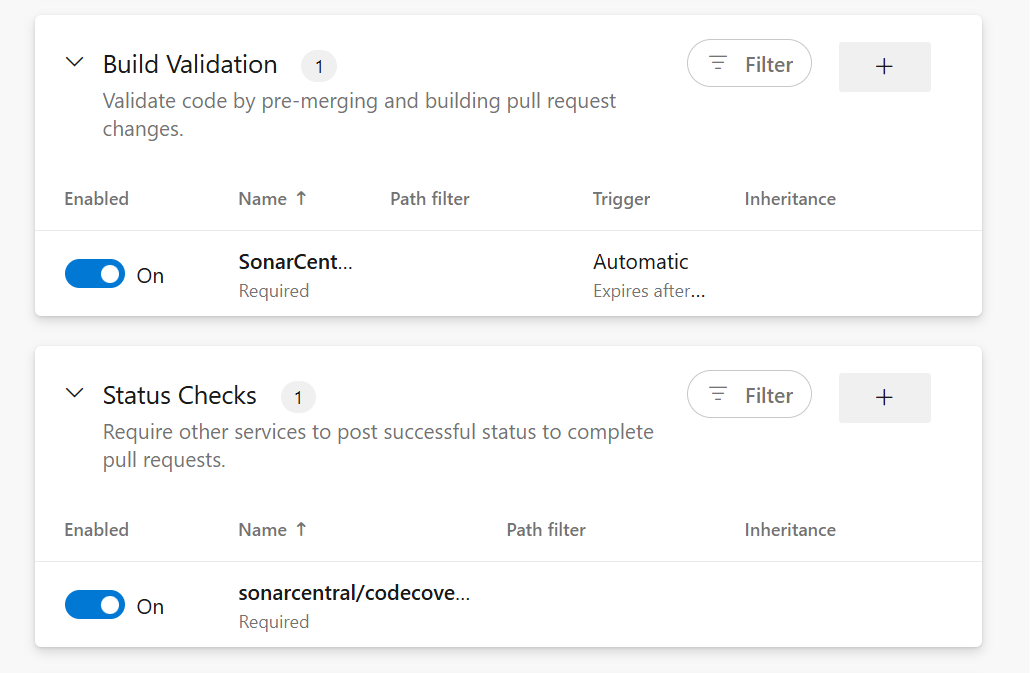
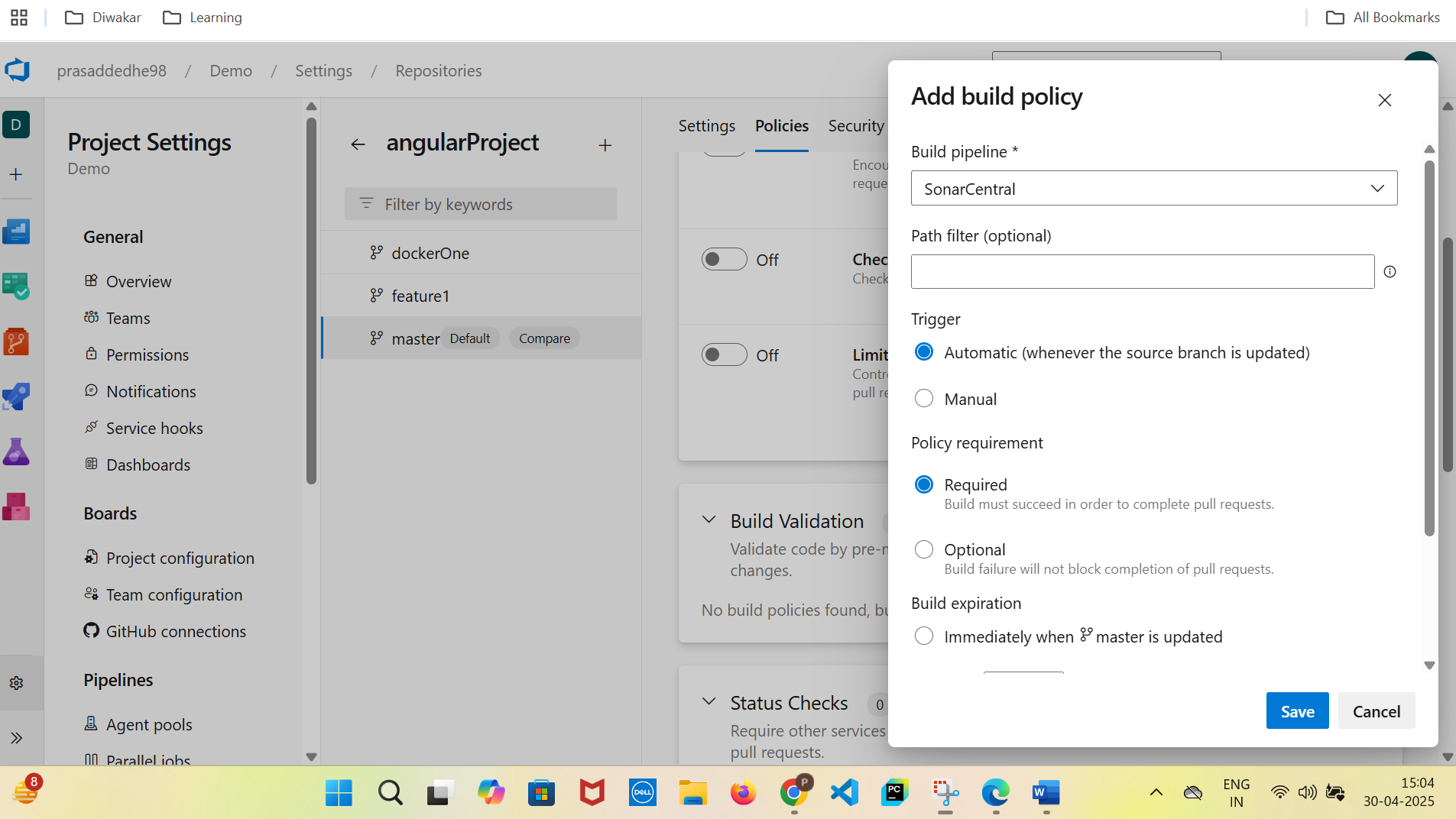
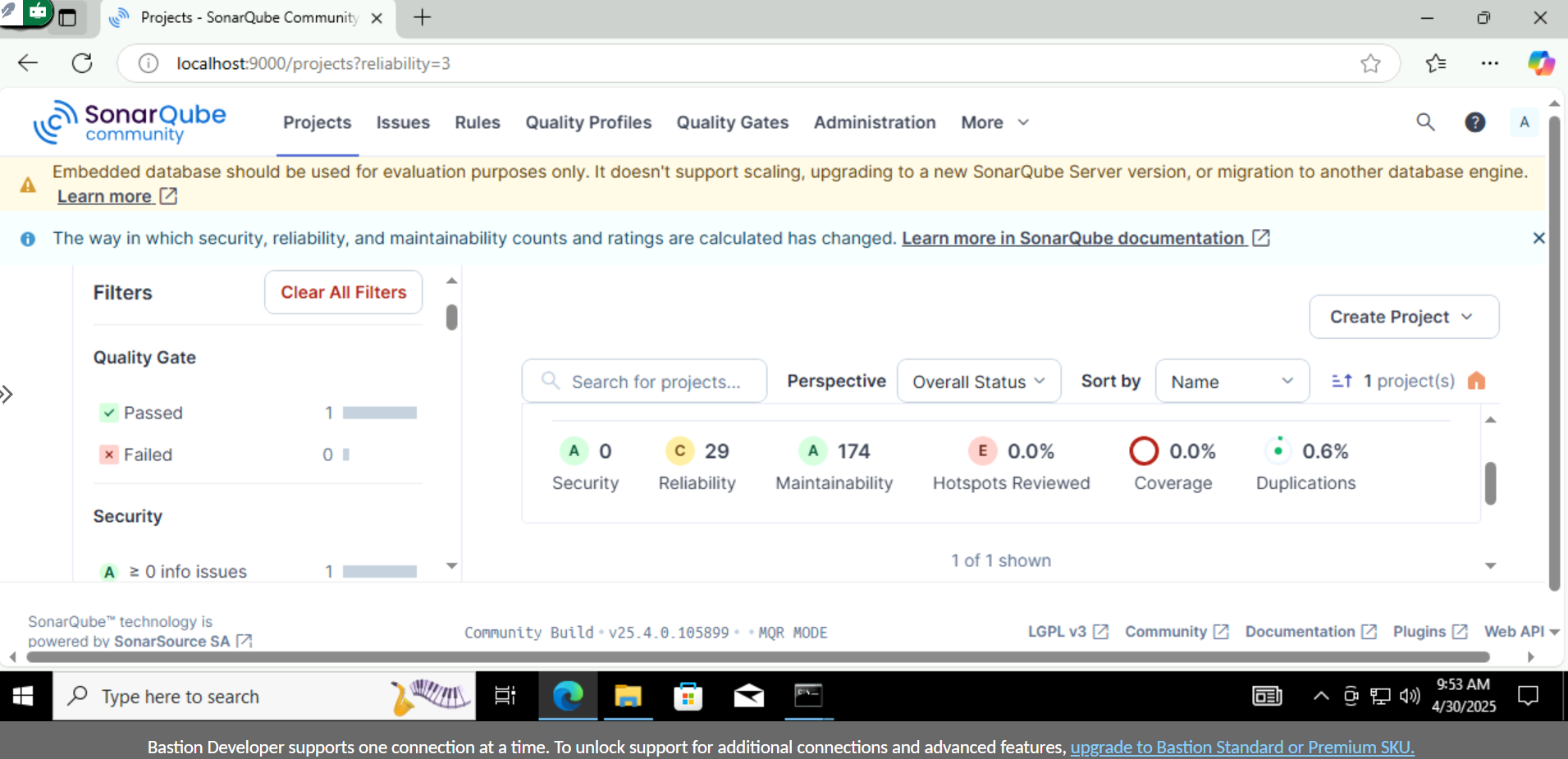
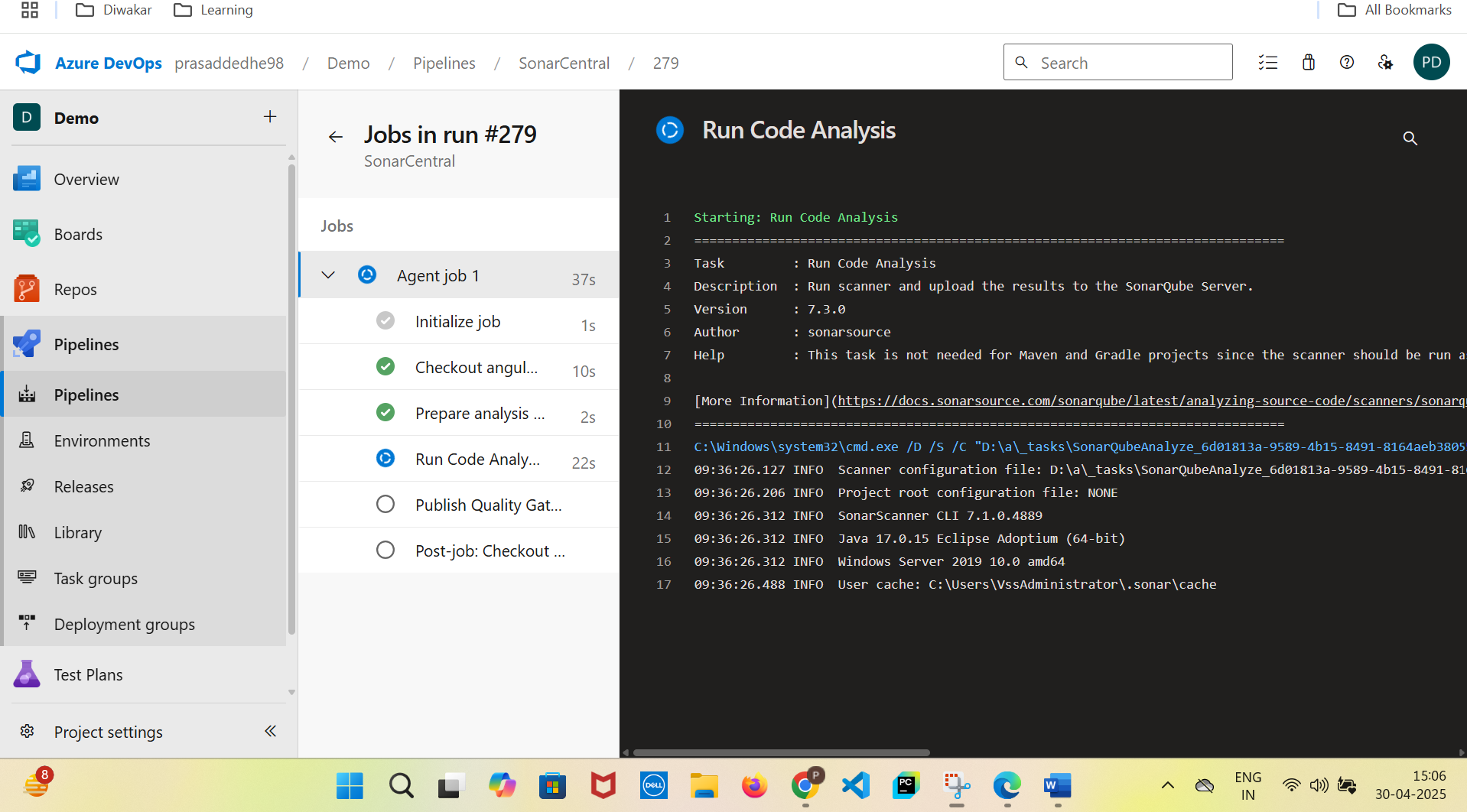
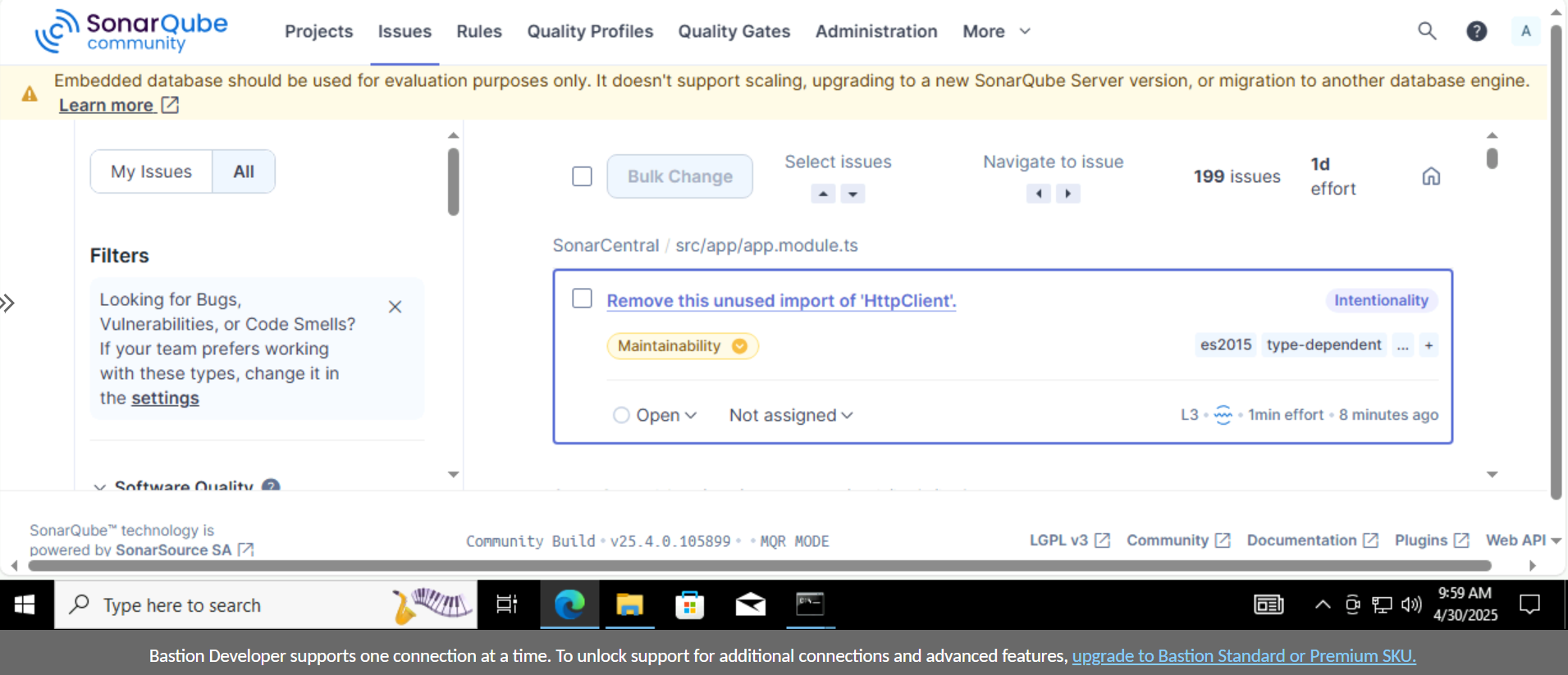
**SonarQube with Azure DevOps PR**

1. Create a VM in Azure. In windows 11 RDP is disabled, so access it using bastion. Keep the size above D else there will be issue accessing it. Once VM is created with NSG allow the RDP rule and login into the same.
2. From google download the SonarQube setup. After that install Java 17 as SonarQube runs on it. Post which add below env variable from PowerShell
   1. setx JAVA\_HOME "C:\Path\To\jdk-17"
   2. setx PATH "%PATH%;C:\Path\To\jdk-17\bin"
3. After this, just run the Sonarstart.sh script (which is in the bin/config folder of the extracted SonarQube setup)
4. Now you will be able to access it on 9000 port. It’s a default port on which you can access SonarQube. After this you can access the SonarQube with credentials (id - admin, password - admin) or URL – http://<public-ip-of-VM>:9000
5. In Azure DevOps market place go and install the SonarQube extension. And once that is done, you would be able to see the tasks if SQ.



1. Create a service Connection in Azure DevOps Portal and add the URL of SQ. Here it will ask the token, so create a local project in SQ and generate a token like below.
2. Once done, you can now create a pipeline with three simple tasks as below. It’s a really simple pipeline. Here you can mention any branch, doesn’t matter, when the PR comes, the build will automatically get triggered as per the branching policy.
3. Now, the main purpose is that whenever a PR is initiated, automatically, the above pipeline should run and the SonarQube should run and then if the result is not good (like there are bugs/code smells) the PR should automatically gets rejected. For this Azure DevOps gives below two options. Build validation and Status check these options can be enabled in the branching policy of the main branch. In Build Validation, basically we add the name of the pipeline so whenever the PR is intiated, then the pipeline is triggered. And status check means, the SQ will send the report back to Azure DevOps, basically if it’s okay or not. As per that, the PR will either pass or fail. (as an example the PR is failed) 
4. This is how the pipeline will look once start running and the SonarQube Dashboard will list all the issues with the code. 



1. Now, the main thing to remember is SonarQube community edition is a waist, it doesn’t allow PR validation (in above ss, the PR validation mainly fails because of that) You have to buy developer edition so it works fine. The process is same.