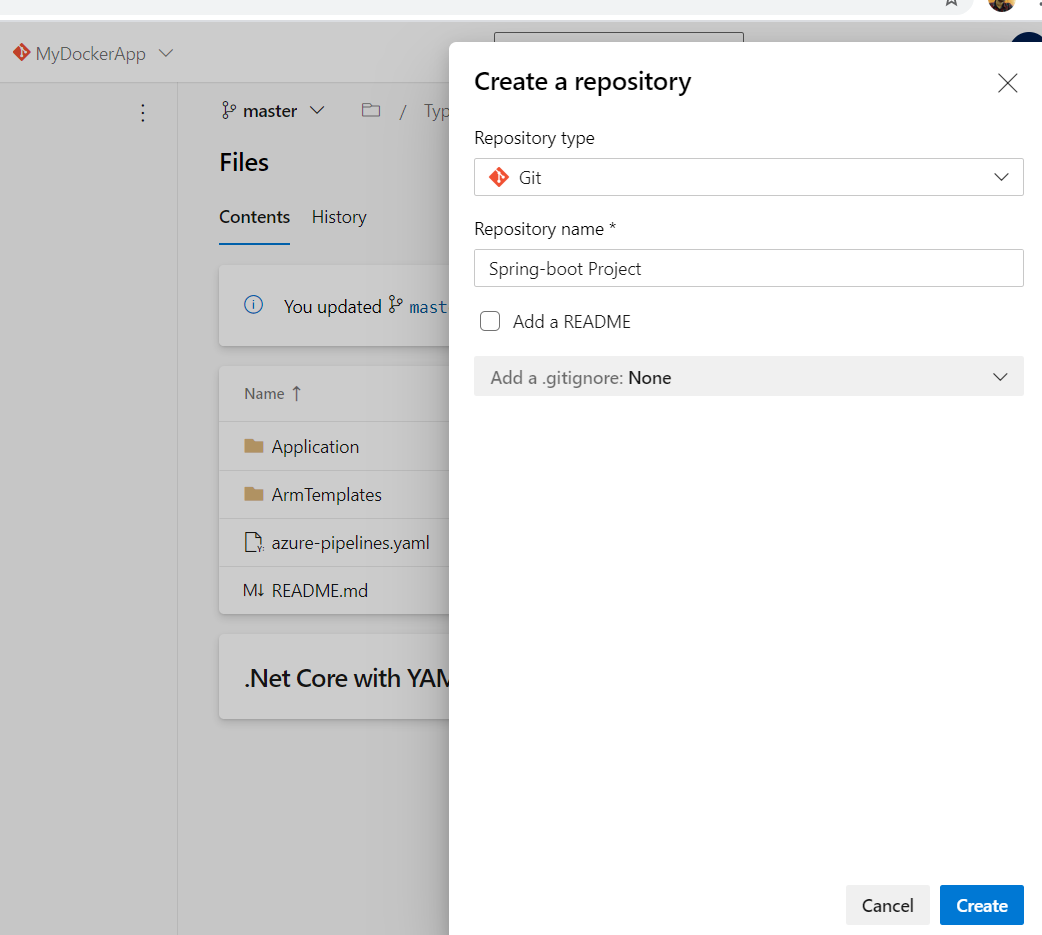
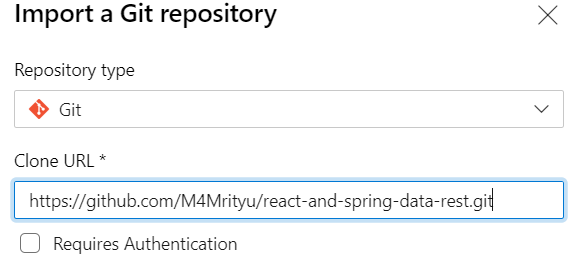
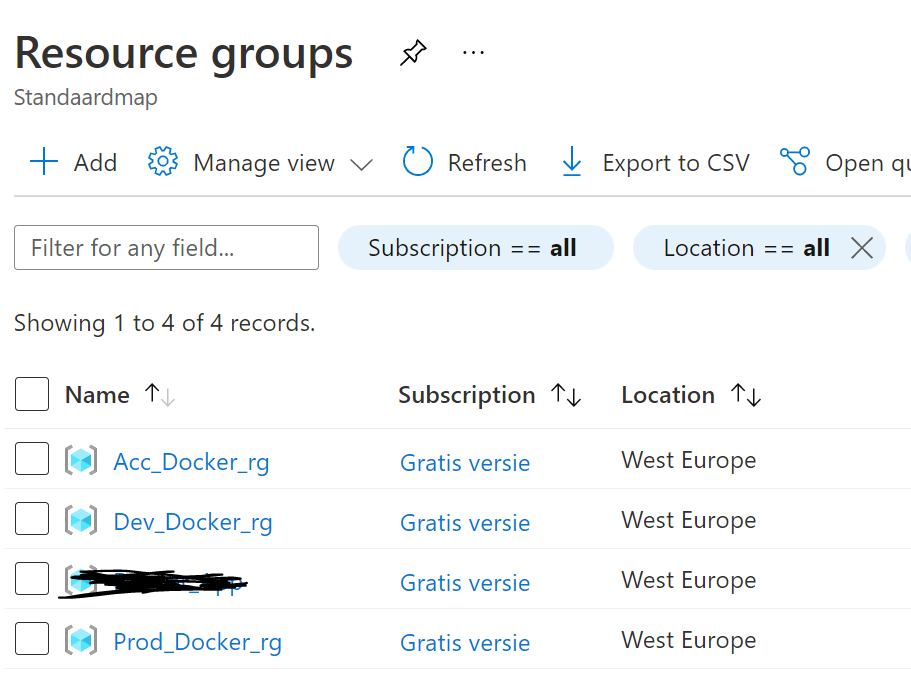
1. Create Git repo in Azure DevOps:



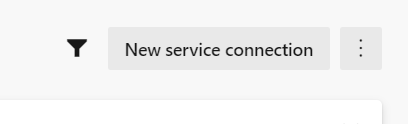
Import Github project. Please use below URL to import the repo.

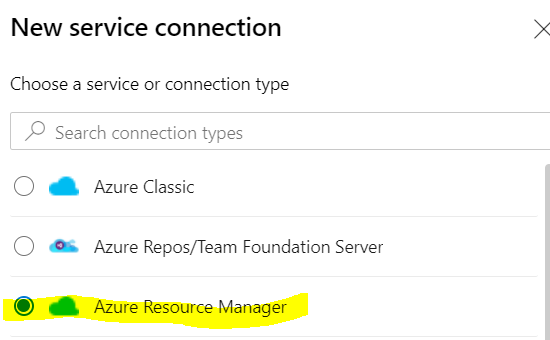
Link: <https://github.com/M4Mrityu/react-and-spring-data-rest.git>



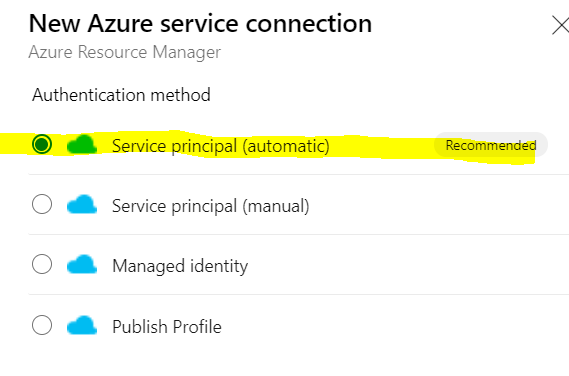
1. Login to <https://portal.azure.com> and create Resource Groups for DEV, TEST and PROD. 
2. Create service connections as below:

a). Open project settings and click on ‘Service connections’ under pipeline section. Now click on ‘New service connection button’ from the right top corner of the page.

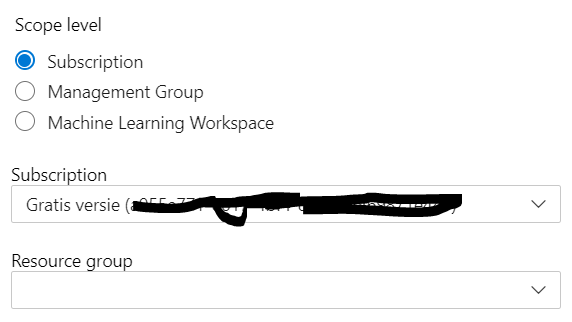


b). Select Azure Resource Manager and hit next.

c). Select Service principal (automatic) if Azure and DevOps account is same. Otherwise, select manually and provide the values.

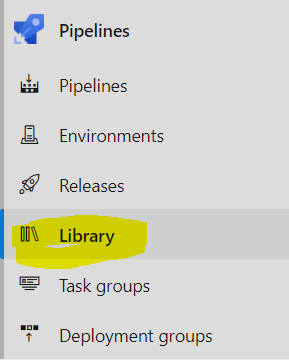


d). In next screen, provide the credential of Azure account. Select the Resource Group and save. Create service connection for all three resource groups.

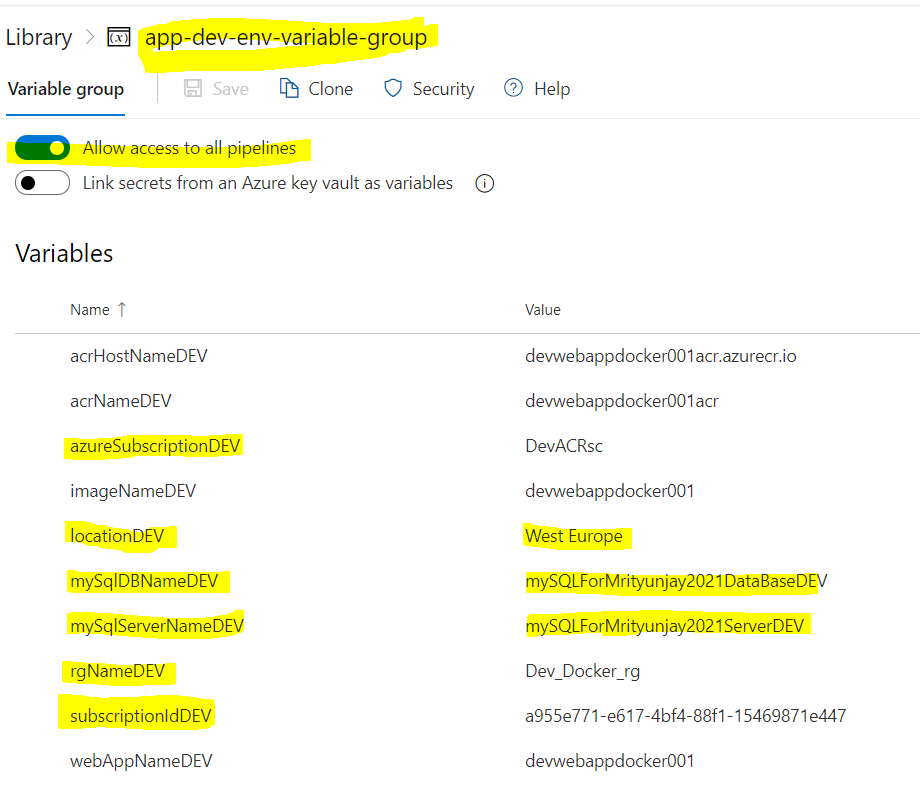


1. Now create Variable Groups to pass the value in YAML file.

a). From pipelines section, click on Library.



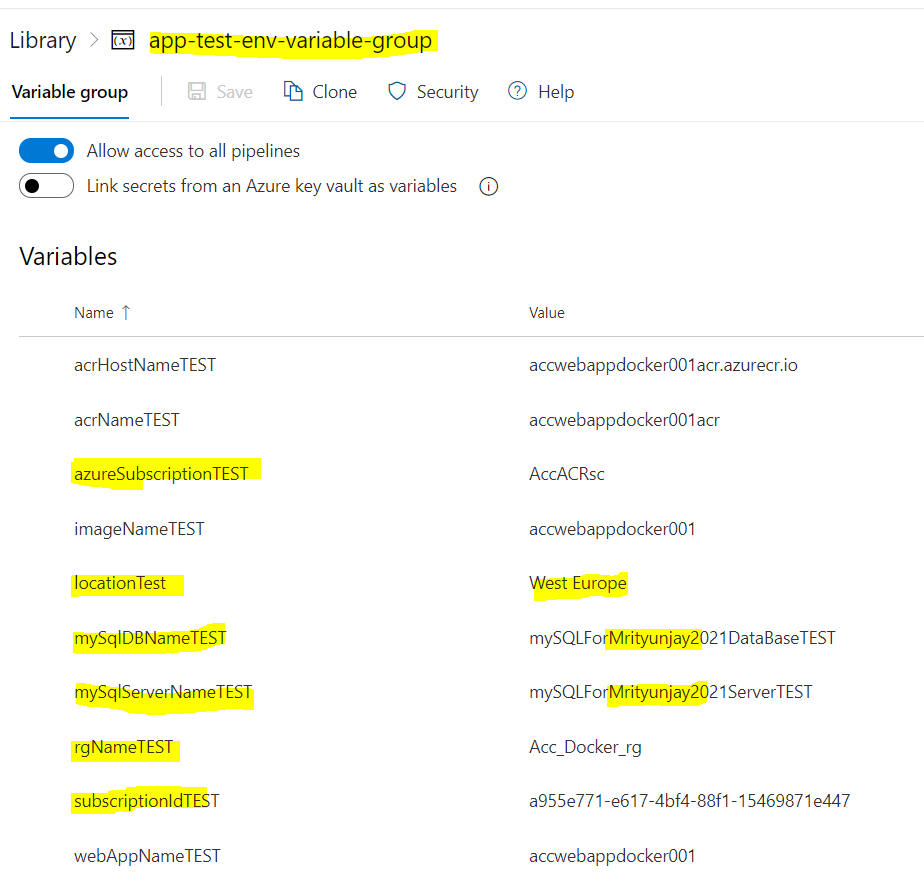
b). Create three Variable Groups for Dev, Test and Prod. Please include below variables inside the group. Let’s begin with Dev



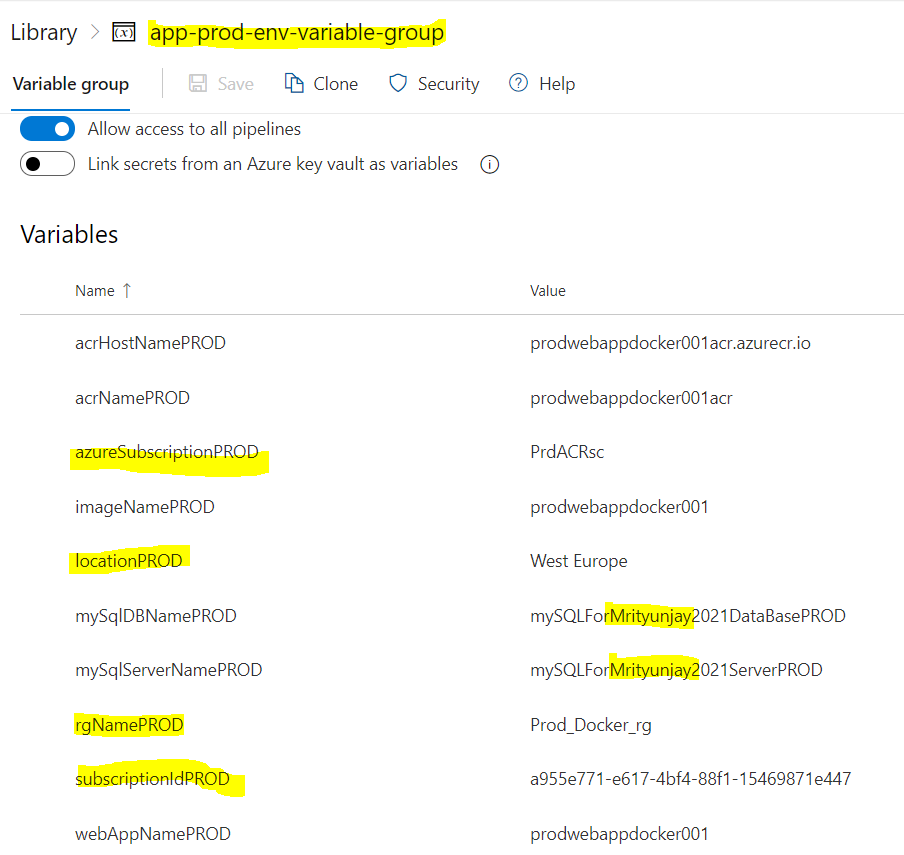
***Note:*** Enter the Service Connection name for **azureSubscriptionDEV** field. And mention your resource group name for **rgNameDEV** field, mention your location for Azure account in **locationDEV** field. Remaining field value can be anything as per your choice.

Repeat these steps for TEST and PROD. It should look like as below:

**Variable group for TEST:** app-test-env-variable-group



**Variable group for PROD:** app-prod-env-variable-group

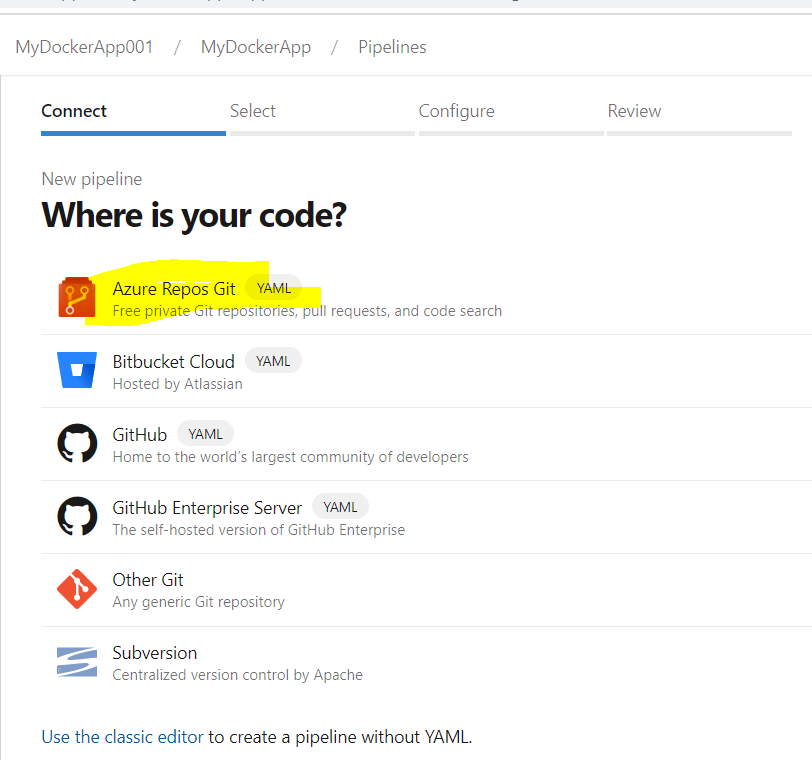


**Disclaimer:** Please make sure that variable groups name and variables name inside the variable groups should intact as mentioned above otherwise pipeline will fail to run.

**Tip:** Create one variable group and clone it. Later edit as per your environments.

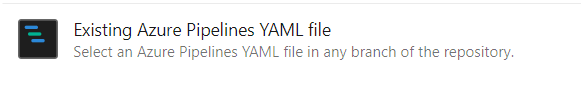
1. **Create Pipeline**

a). Select ‘Azure Repos Git’

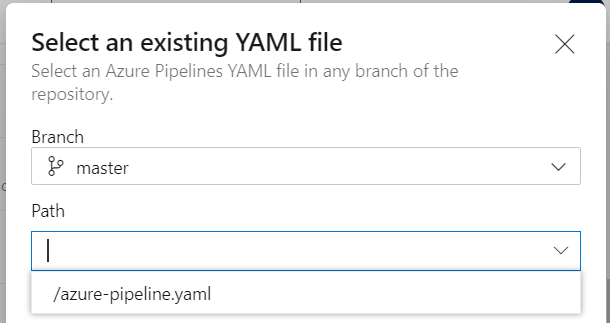


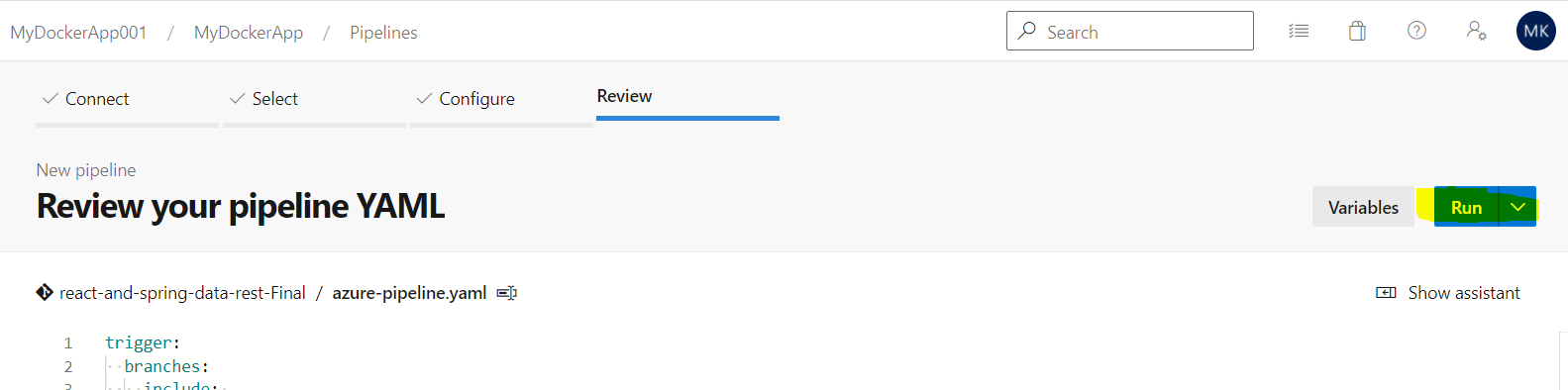
b) In next step please select your repo.

c) Select ‘Existing Azure Pipelines YAML File’.



d) Choose YAML file from the dropdown and hit Continue button.

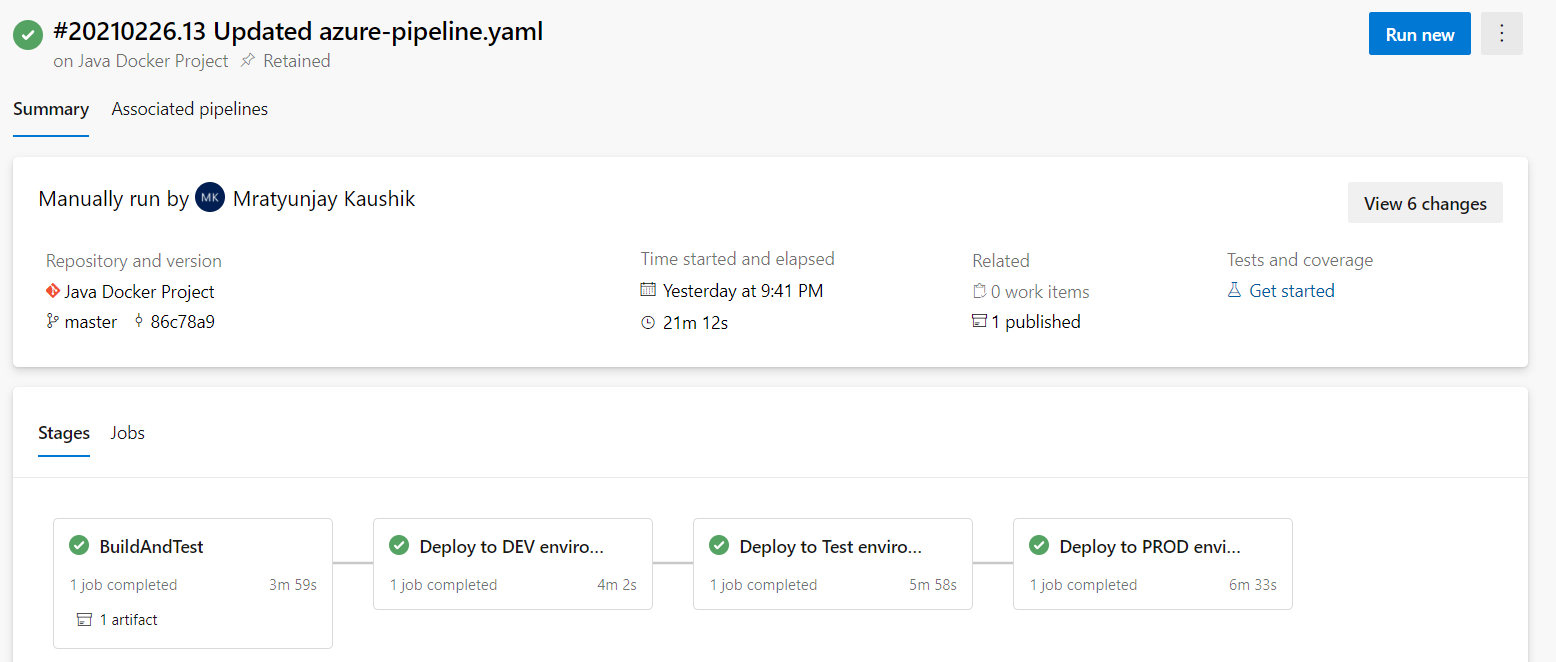




Once you hit on ‘Run’ button the pipeline will be triggered. We can see run from all runs section.

Now developer can commit the changes in the code it will trigger the pipeline automatically. Or also can fire pipeline manually from the UI.

**Note:** I am using Microsoft hosted Agent.



**Note:** *As* ***no unit test case project*** *found in the shared assignment, it skips unit test run in pipeline. However, snippet has been added inside YAML pipeline, but it is commented out.*

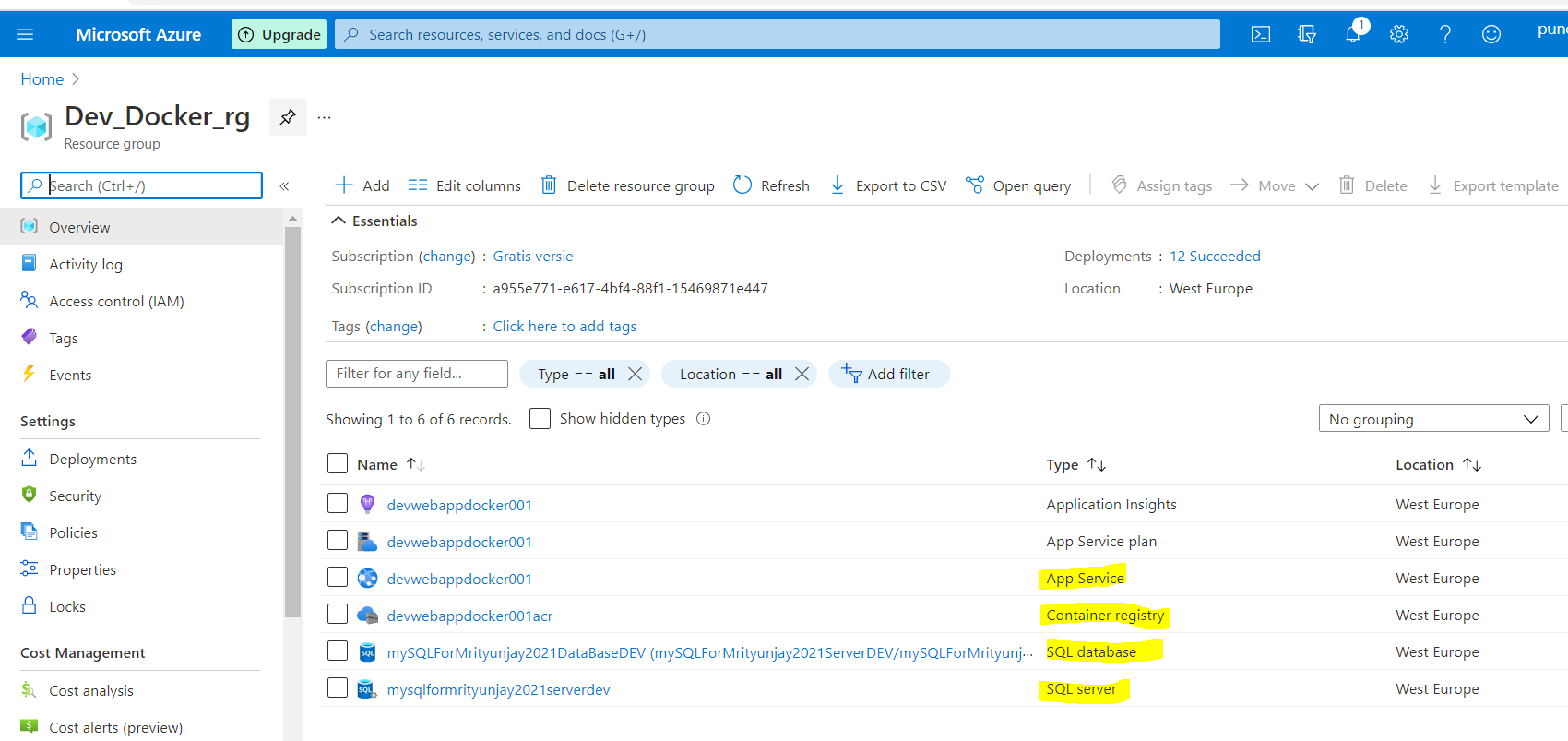


Once pipeline successfully run, it will create or update Azure resources **App Services, Container Registry, Application Insight and My SQL database** as a part of infrastructure. Afterward, our application will be hosted on App Services.

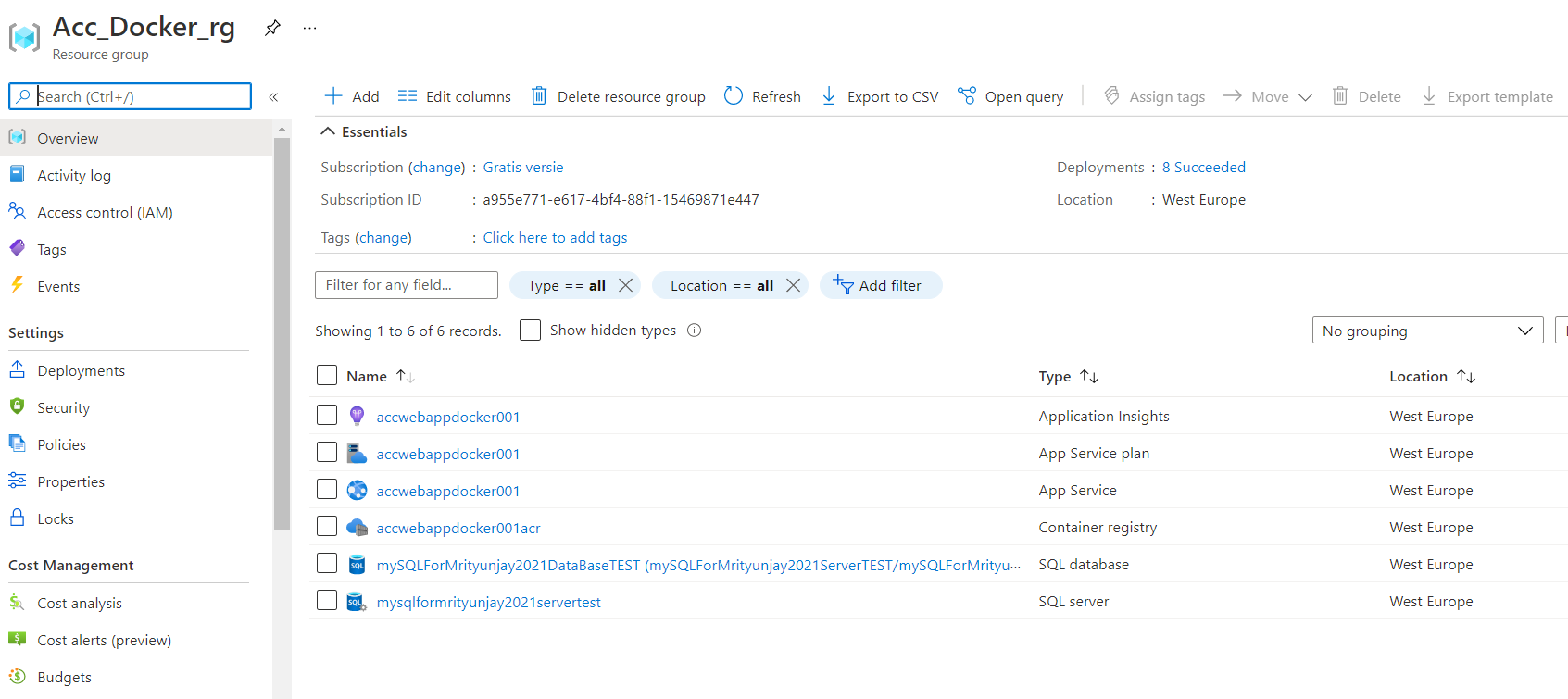
Credentials for databases created under resource groups will be SQLadmin/Myappjava@2021

**Note:** However, database is created but not attached with the application for the use. A Java development background is required for that.

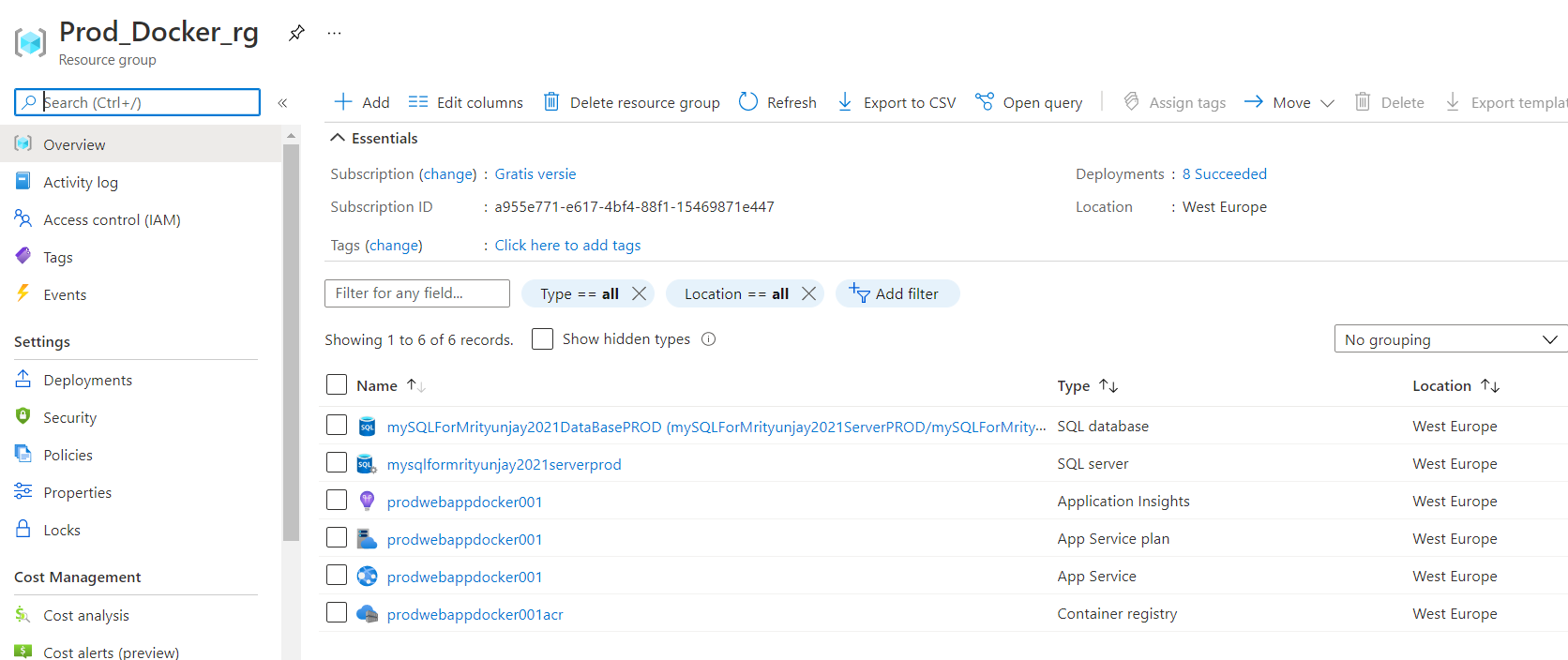
**Dev Azure Infrastructures:**



**TEST/Acceptance Azure Infrastructures:**

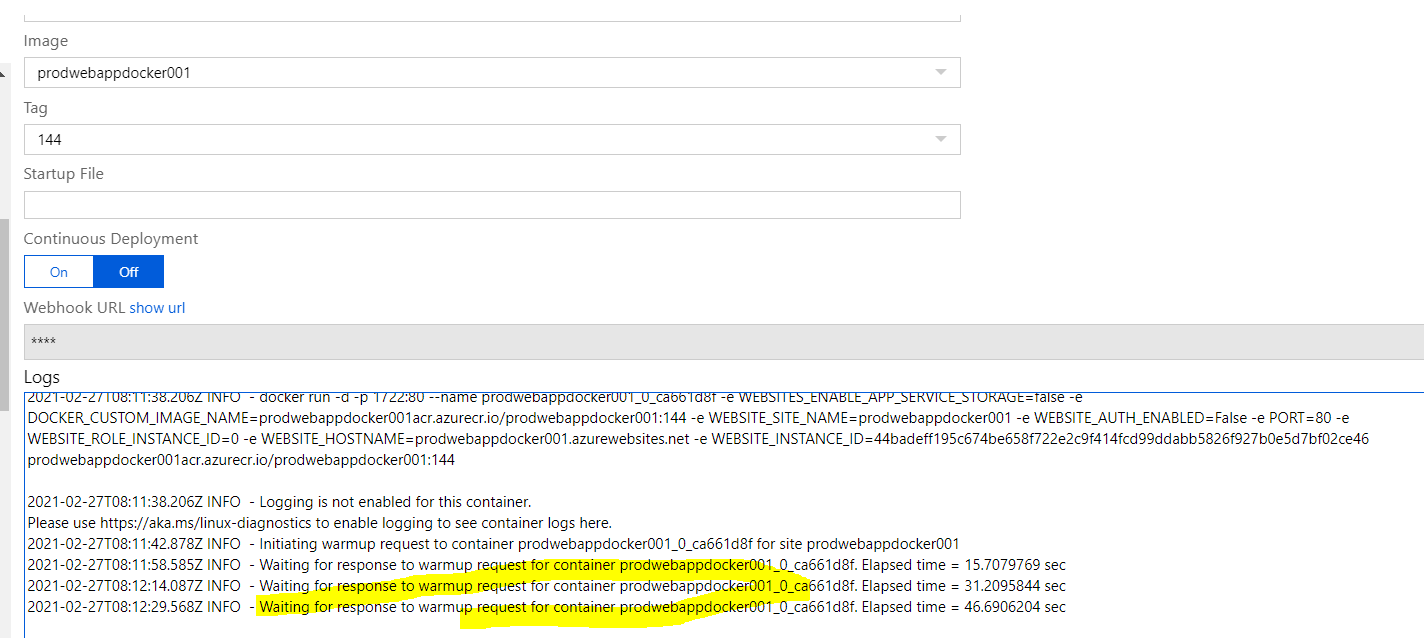


**Production Azure Infrastructures:**



Now, click on App Service to see if application is hosted or not. Click on browser to see application is getting launched or not.

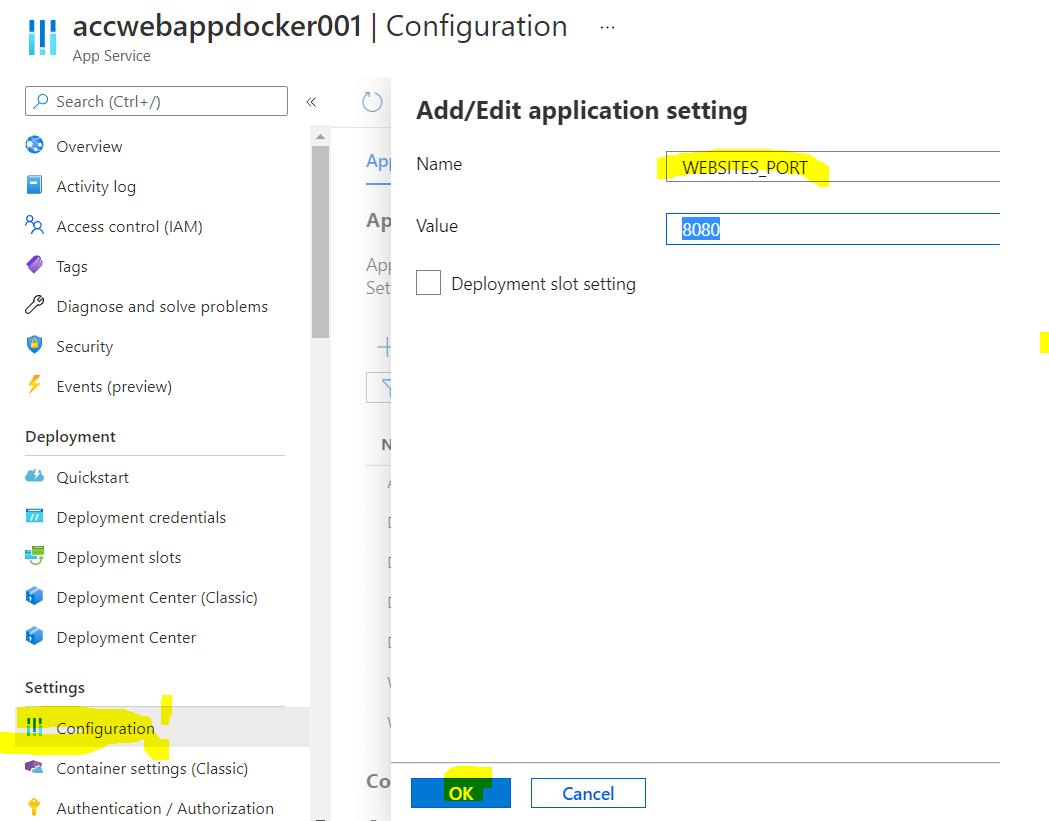
*If it is kept loading, then click on container settings under the settings option you may see below log*.

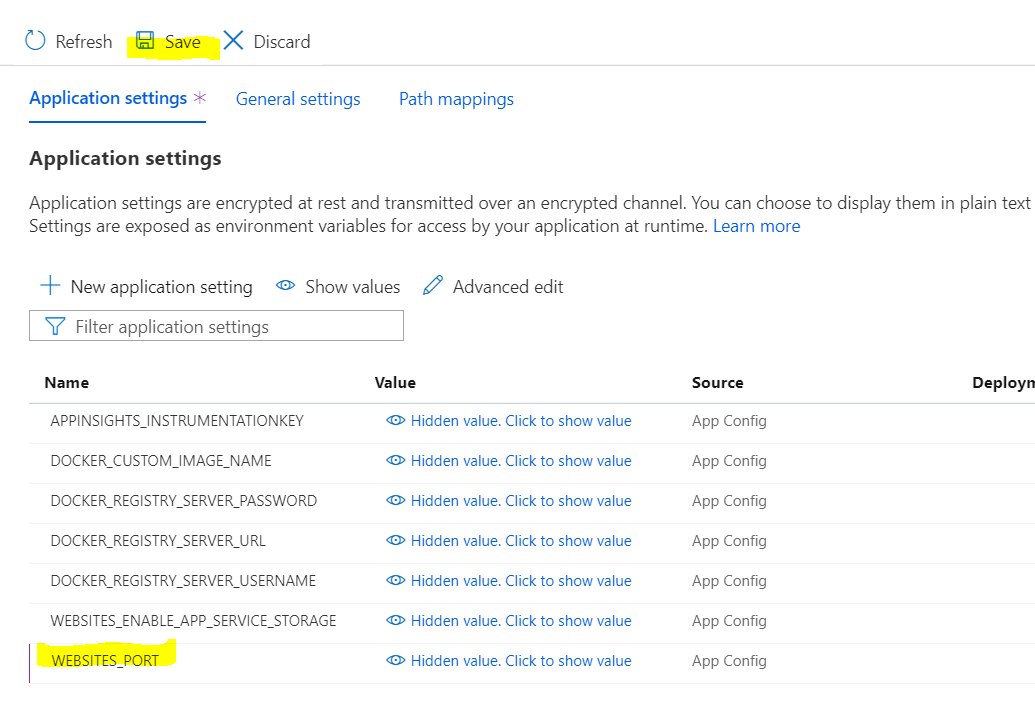


I have verified *.Net core application* that was working fine but *Maven/Java application, on which I am working for first time in this technology,* seems it has some configuration related issue.

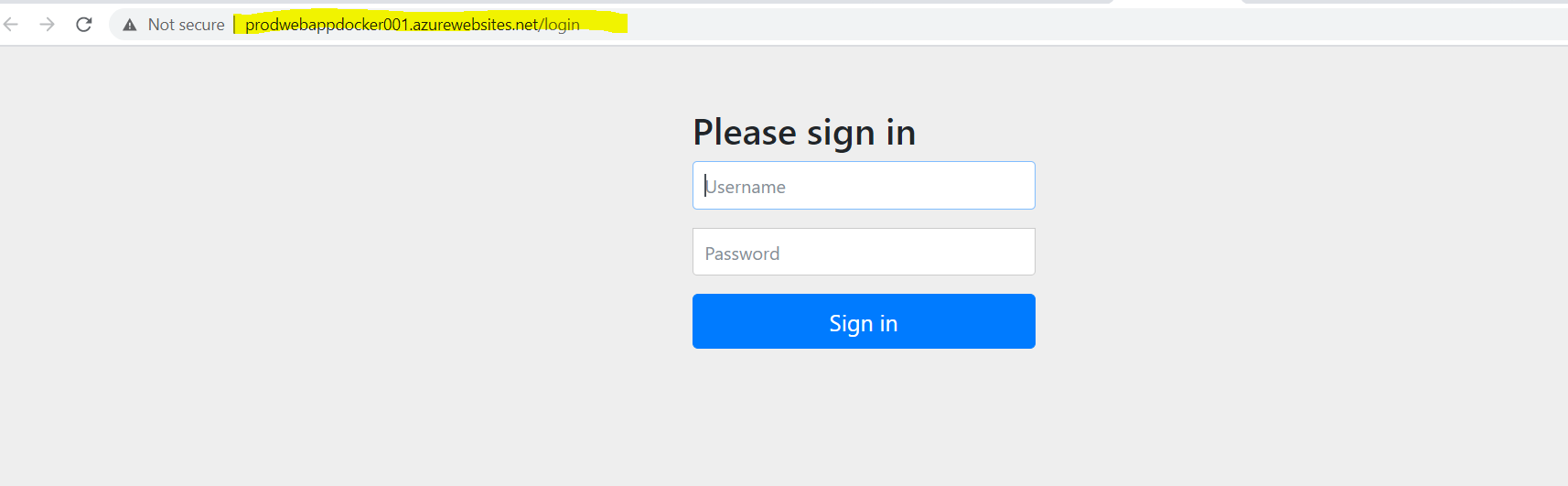
However, I tried to fix it and got some solution. We need to configure below setting.

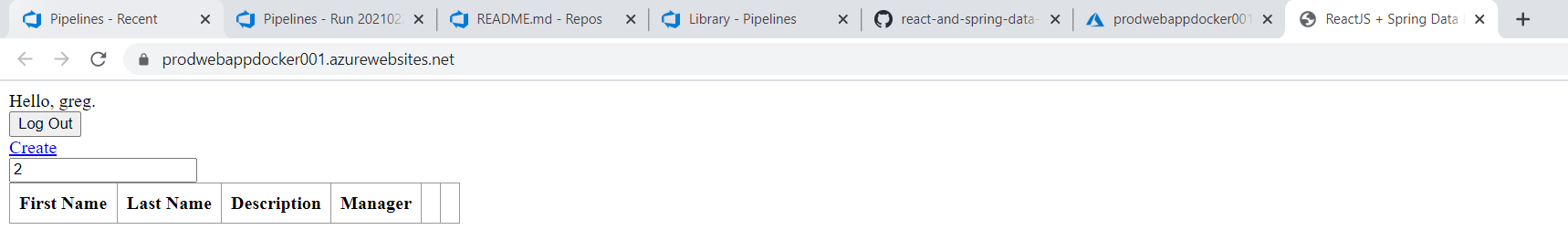
Open Configuration from Settings area and configure WEBSITES\_PORT as below.





Restart the app service and wait for 10 minutes. Click on browse it will open the Maven application.





We have to configure same settings for all three environments. As, all three cannot be run together because of application using same port.

You need to delete the configuration for other App Service for other two environment and restart the App Service that you want to run, once it is restarted browse the application it will work or you can also try with your custom port but in my case it was not working.

Here is reference from Microsoft.

