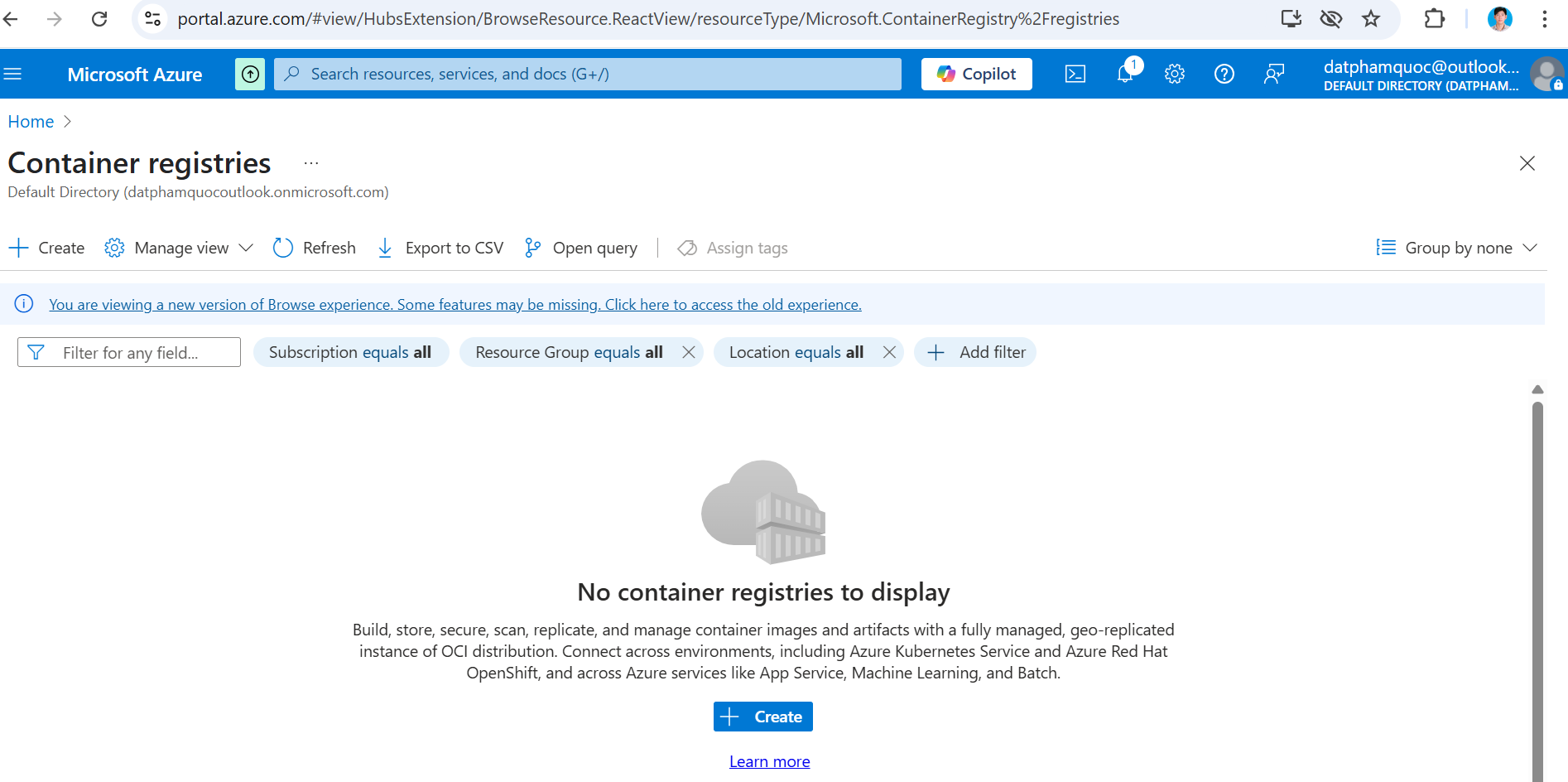
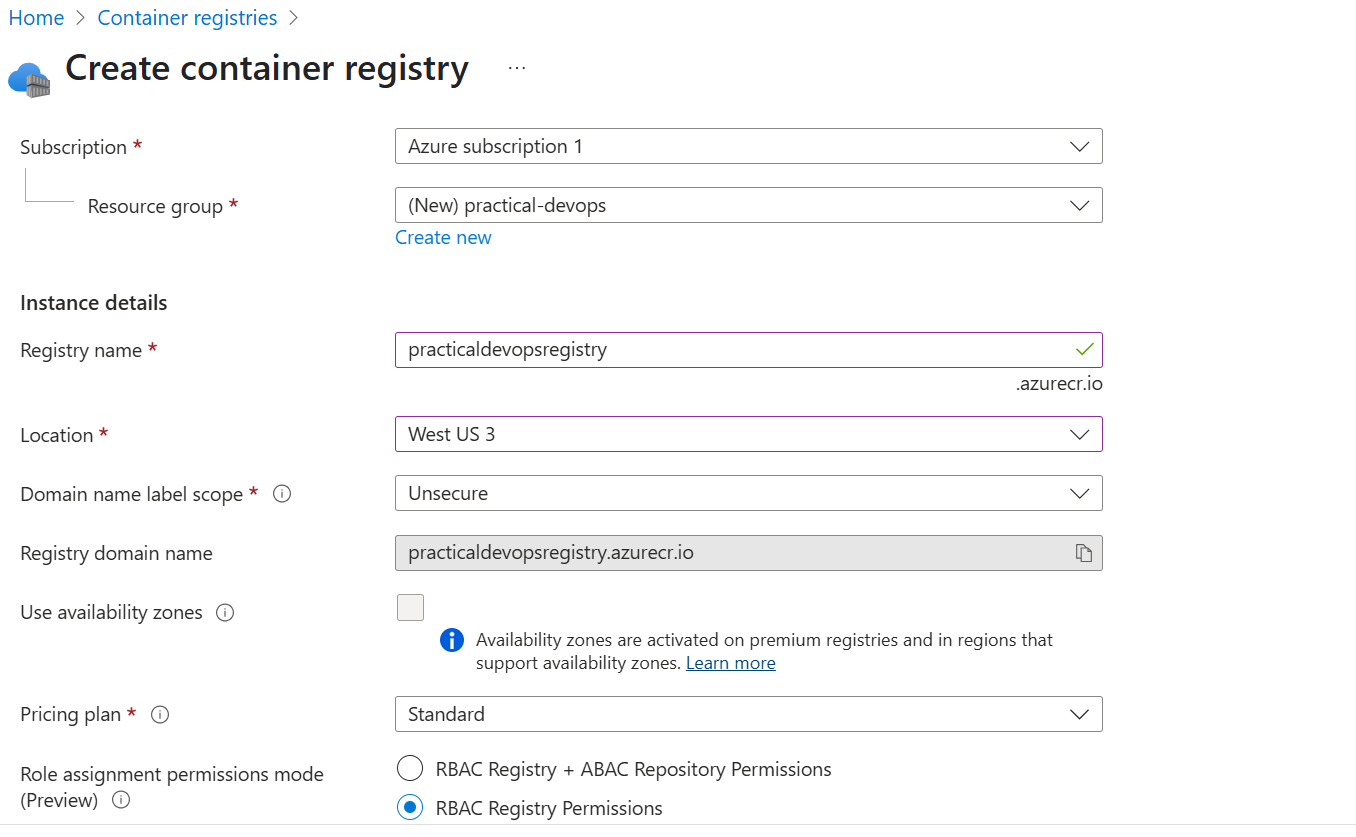
# Create ACR and setting Azure Devops

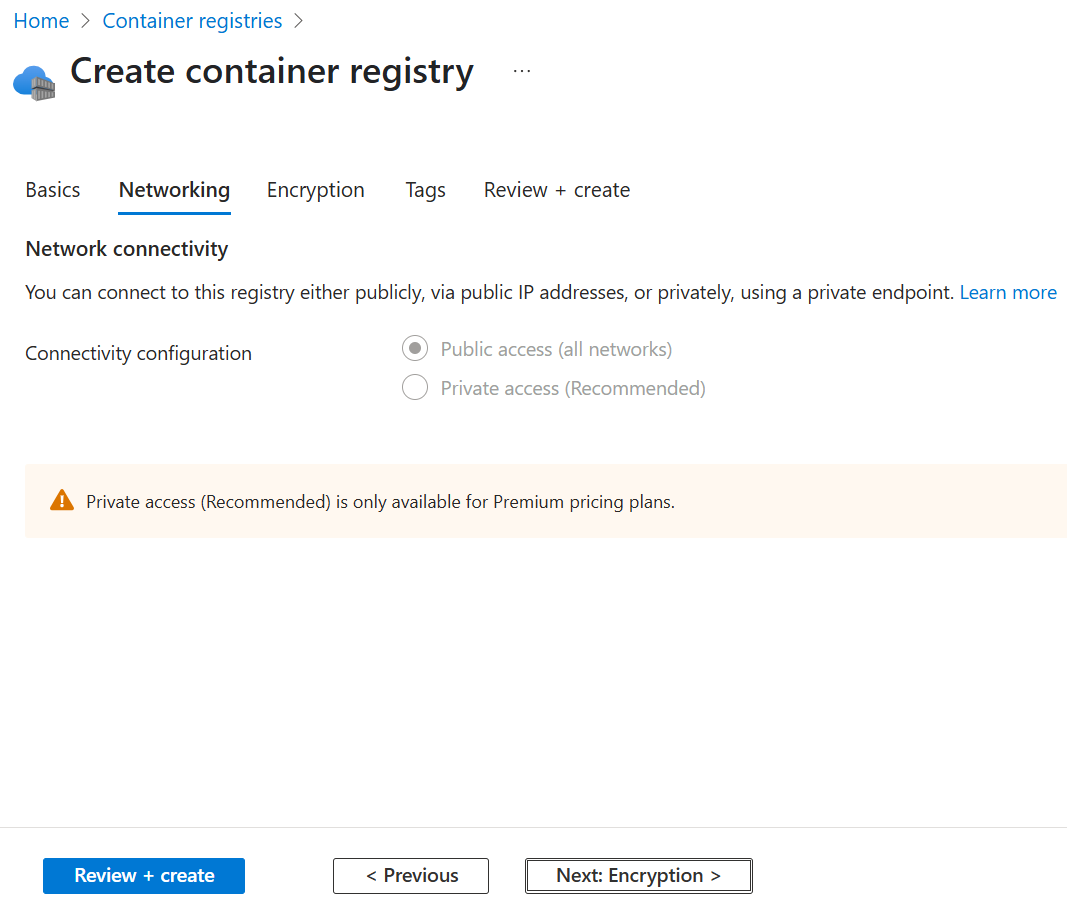
# Navigate to Container registries page: <https://portal.azure.com/#browse/Microsoft.ContainerRegistry%2Fregistries>



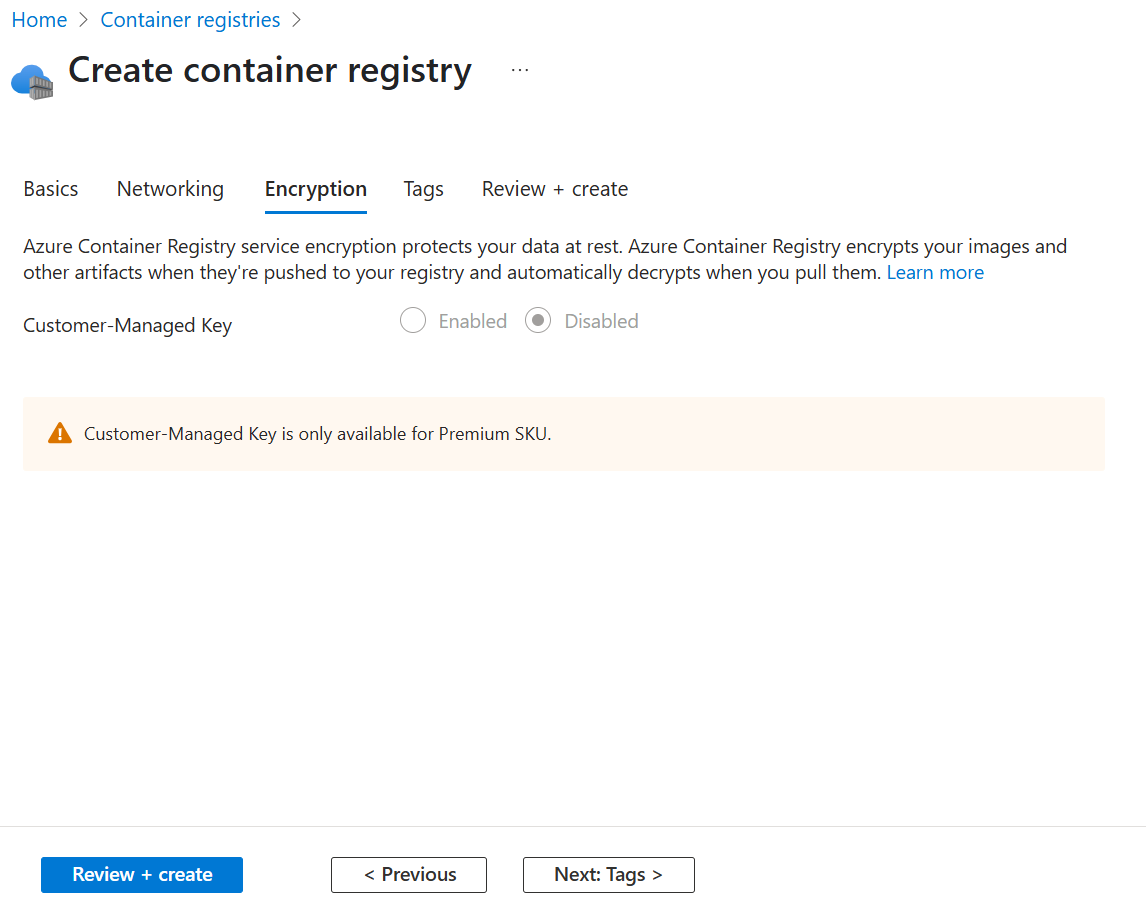
# Click Create button, then put information as below



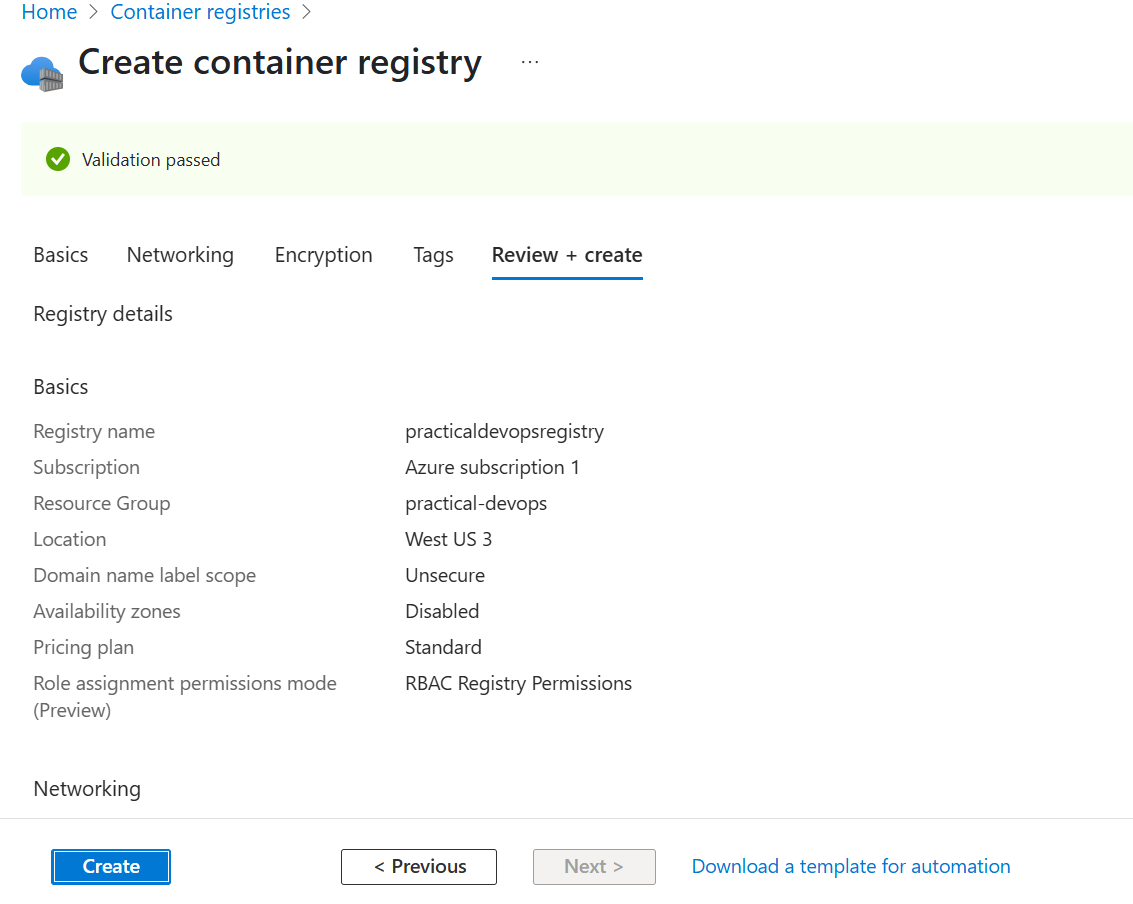
# Click Next: Networking button and keep default



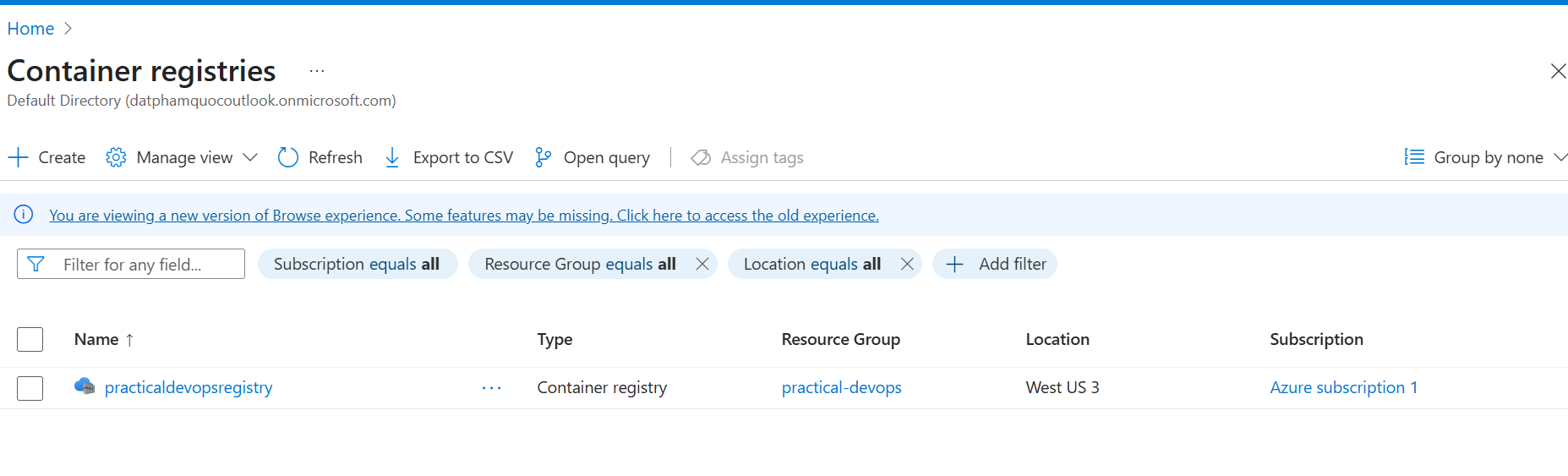
# Click Next: Encryption button and keep default



# Click Review + create button

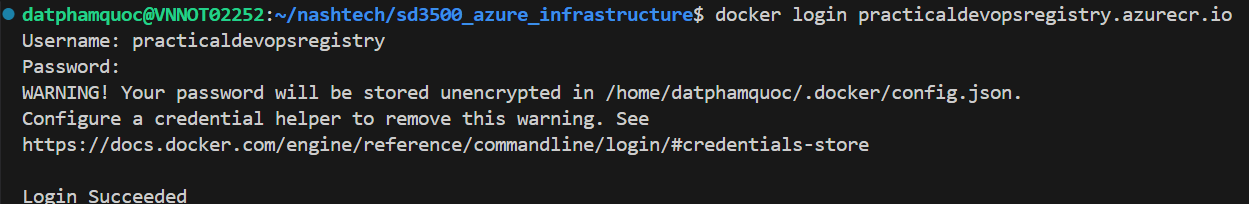


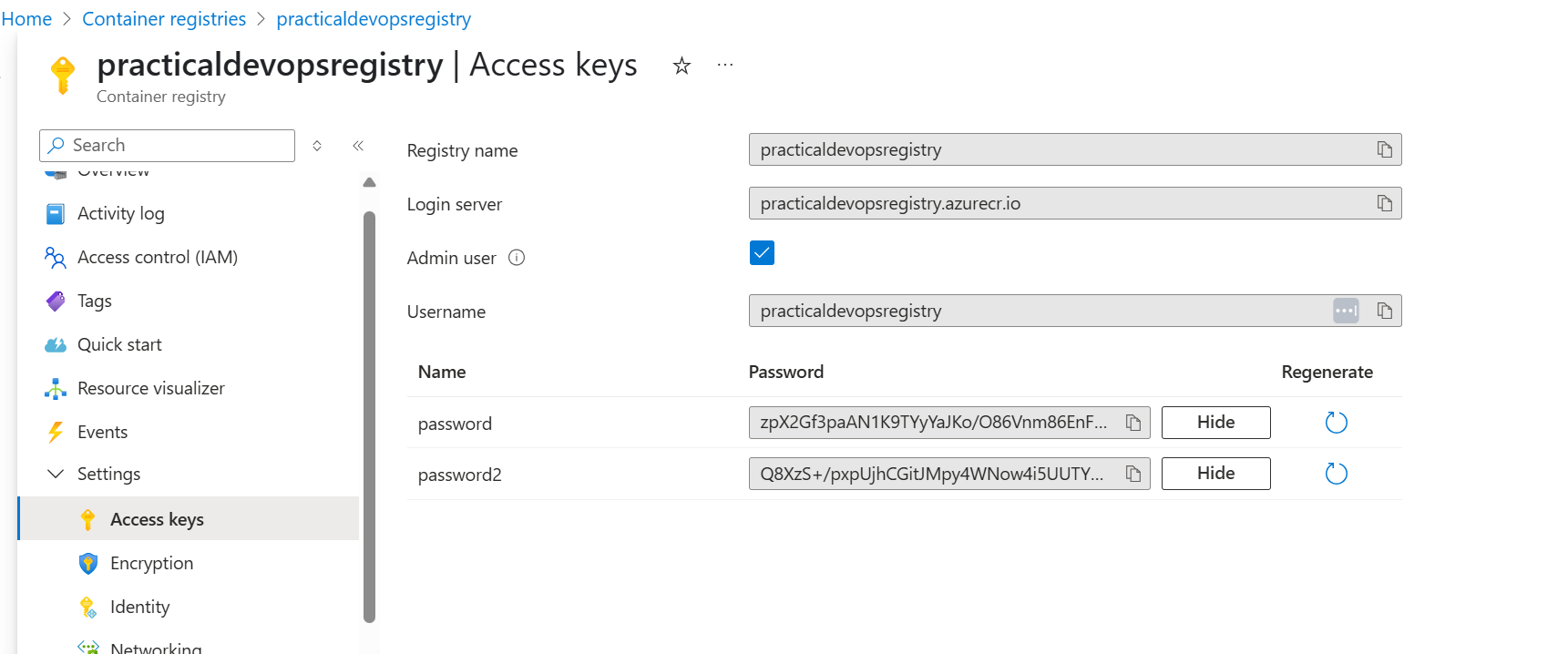
# Click Create, new registry is created



**Login to ECR**

docker login practicaldevopsregistry.azurecr.io



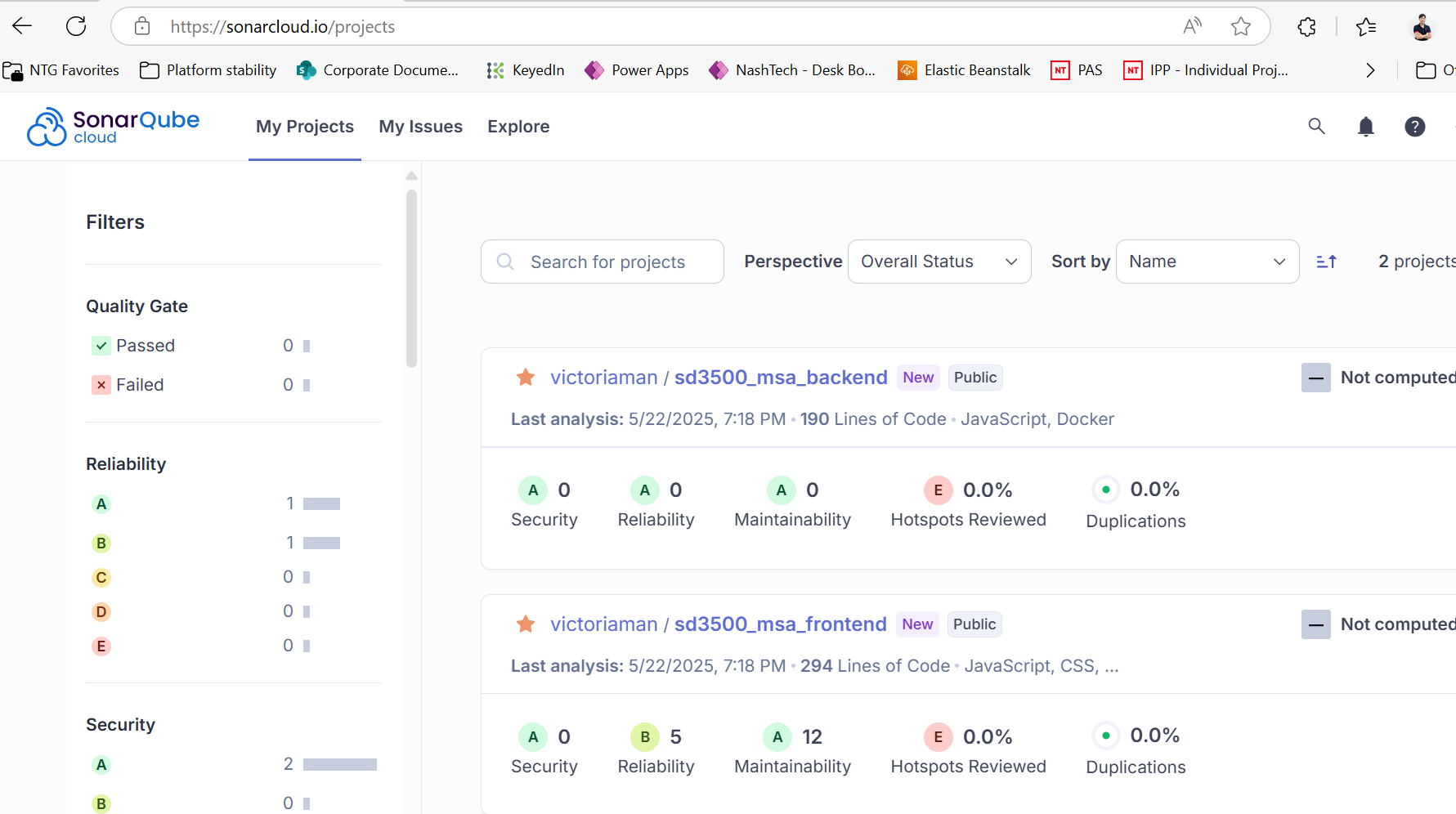


# Configure Azure Devops

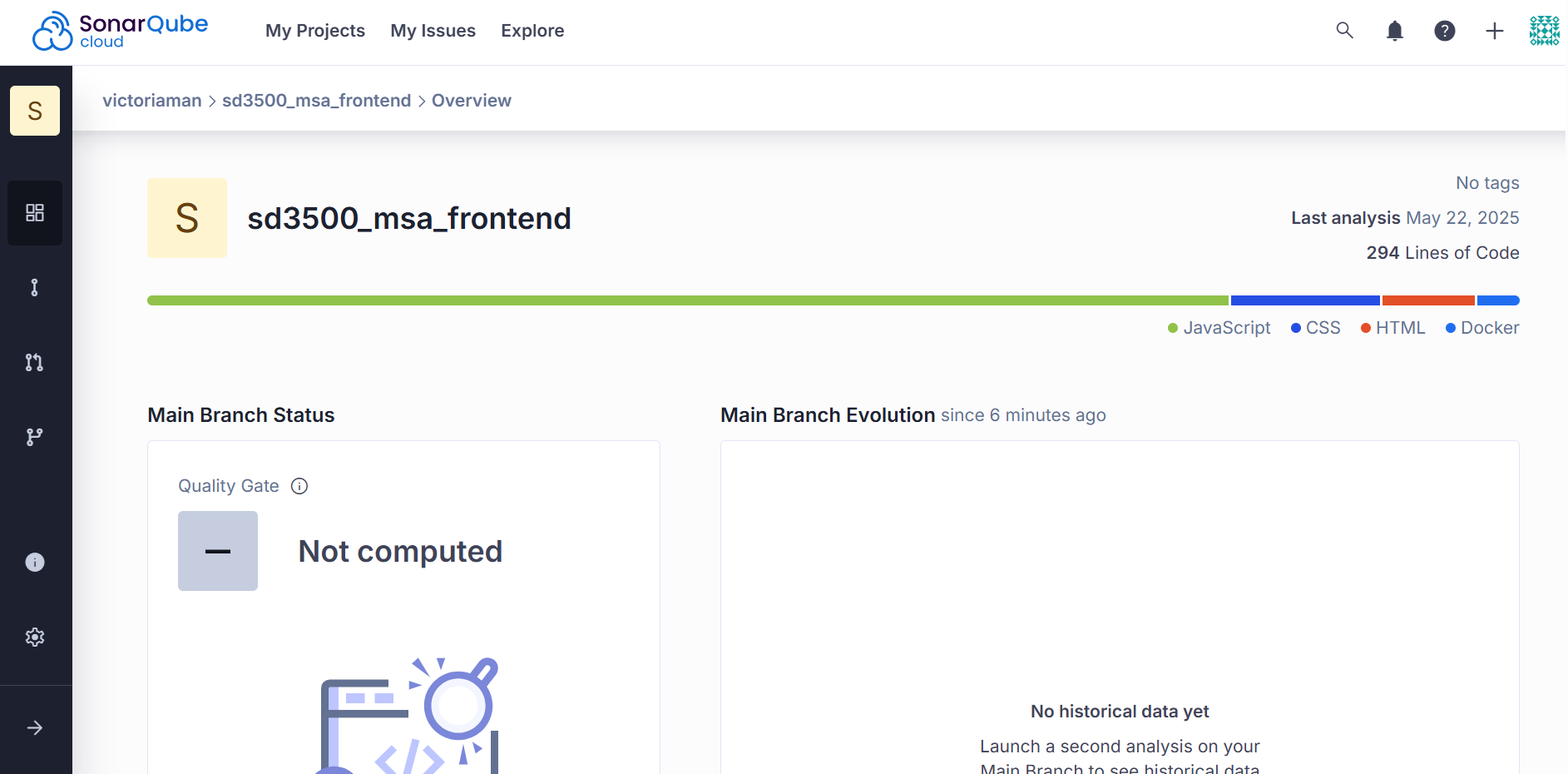
**Integrate with Sonarcloud**

# Create project in sonarcloud

After create account from Github, you can create project for existing repository in your Github.



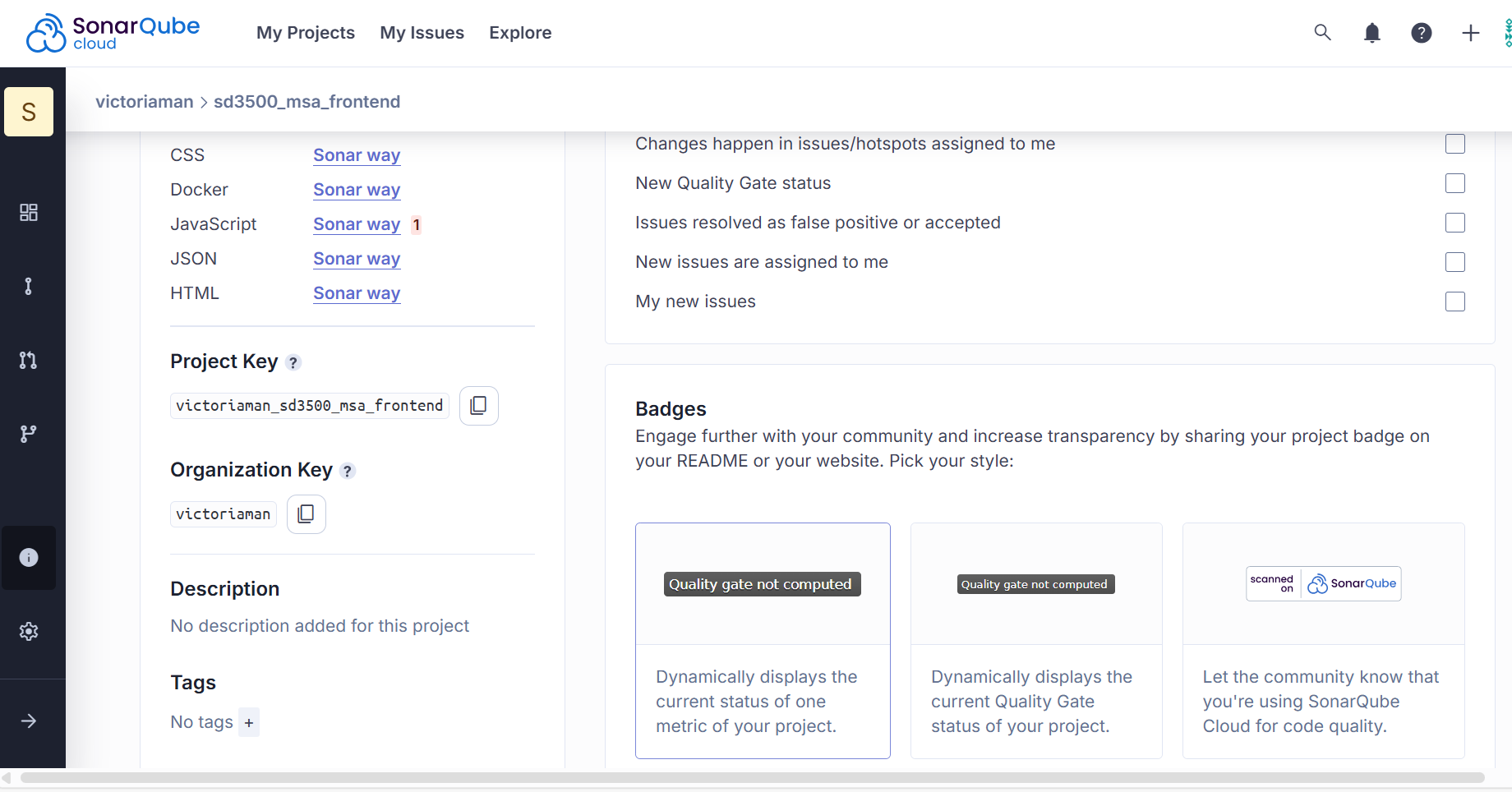
# To integrate Azure Devops with sonarcloud, you need to get project key of specific project in sonarcloud, Organization Key and Access Token

# Choose sd3500\_msa\_frontend project

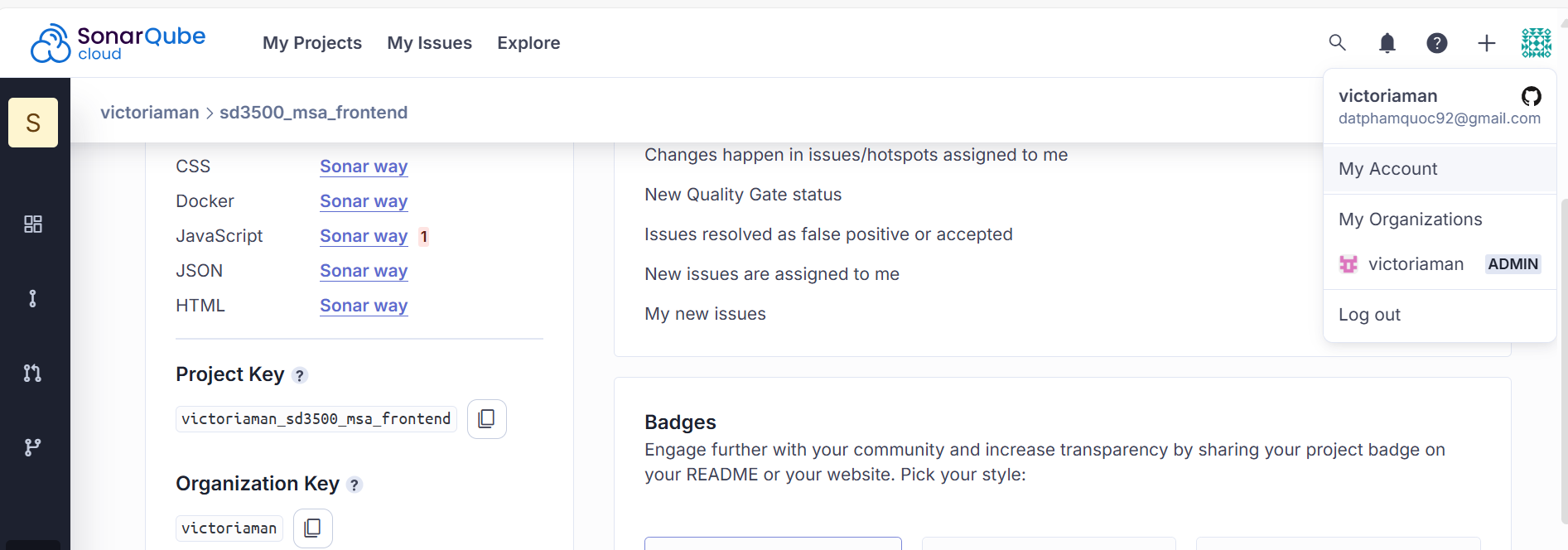
# Click Information menu



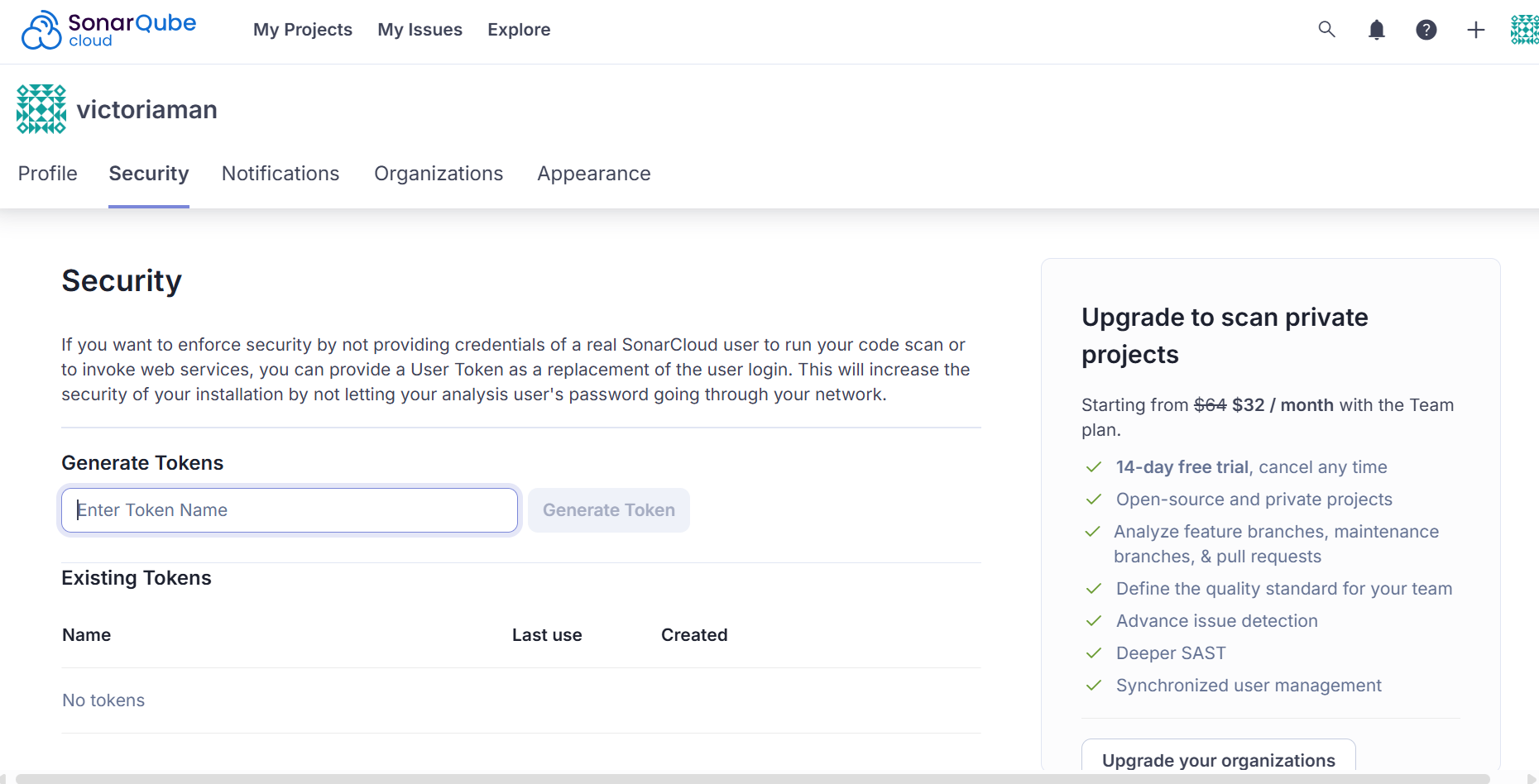
# You can see Project Key and Organization Key



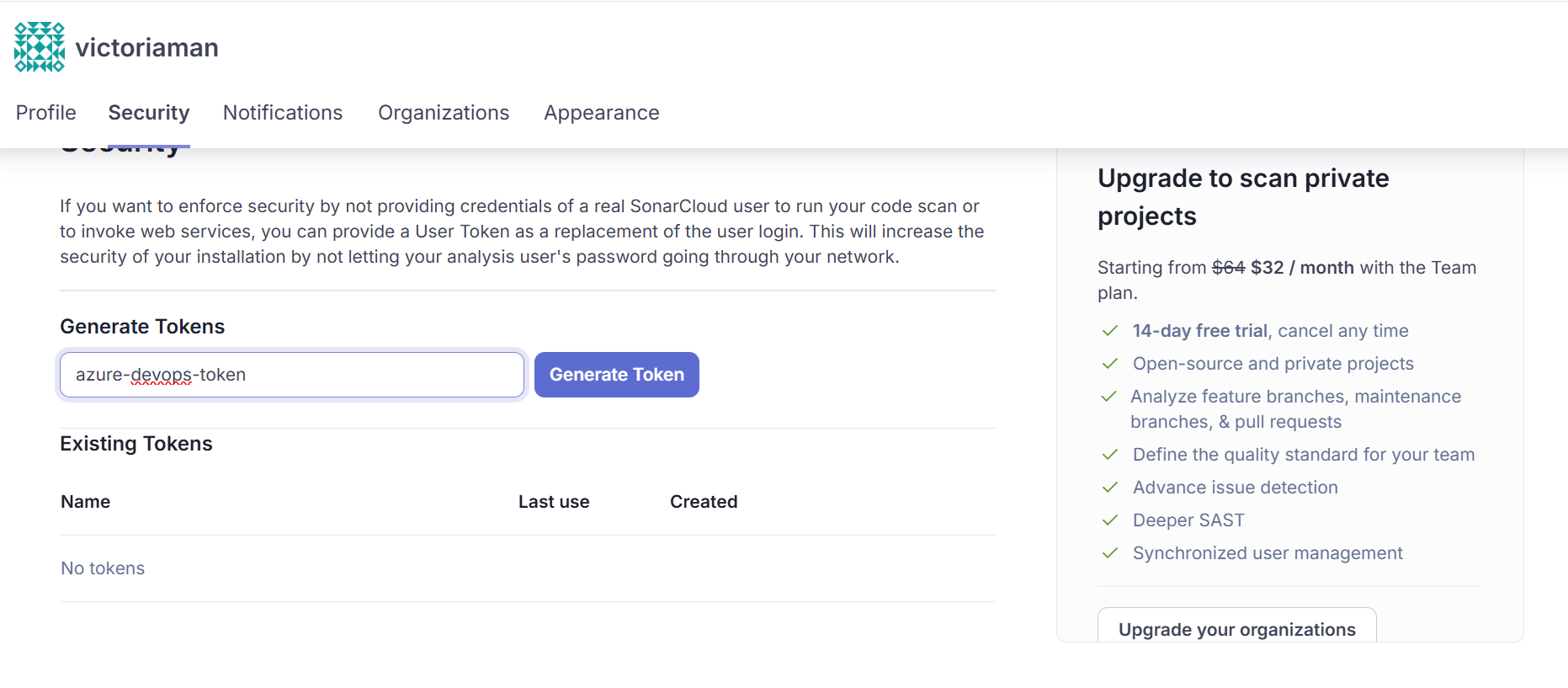
Click My Account

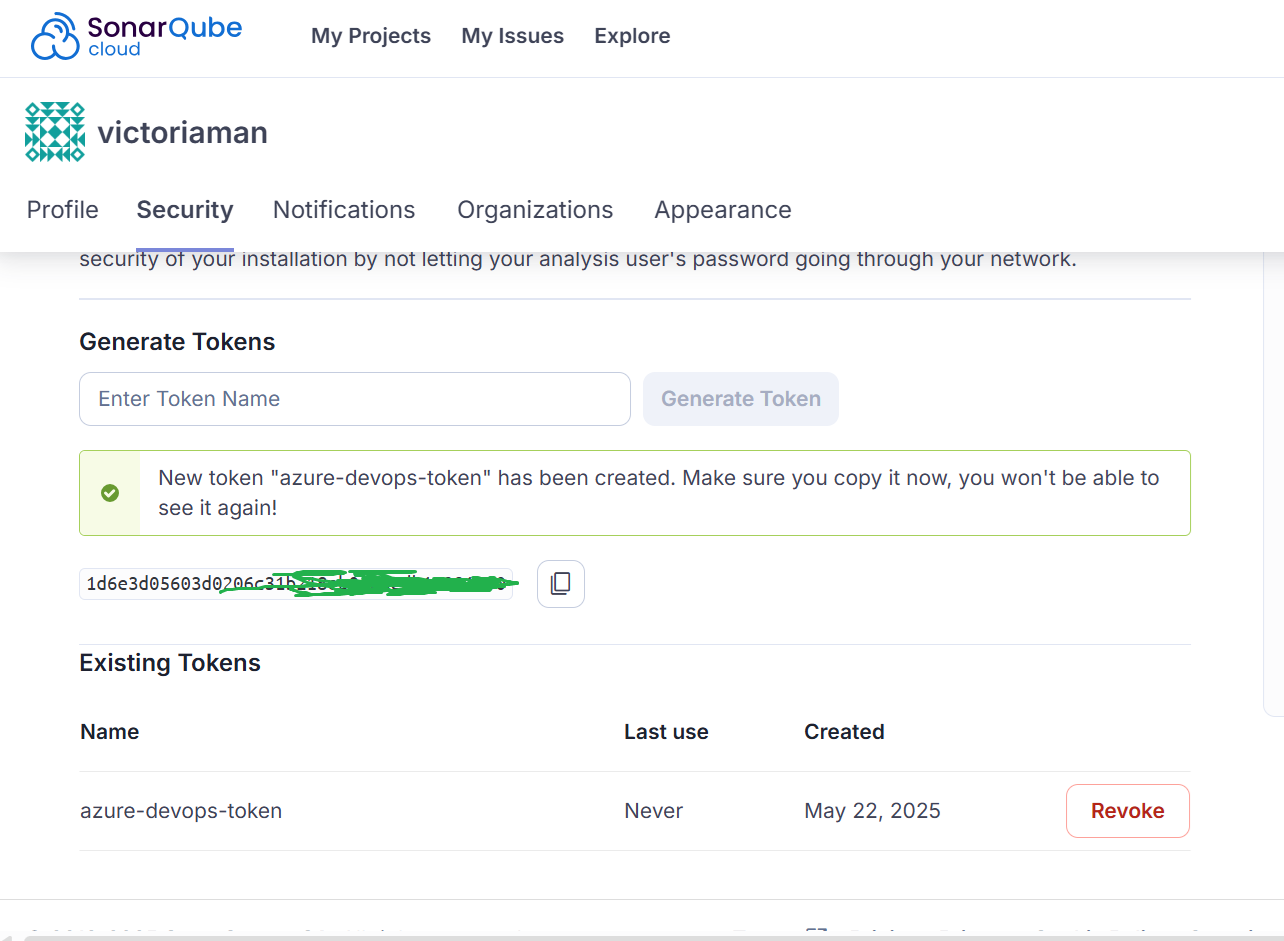


# Click Security tab

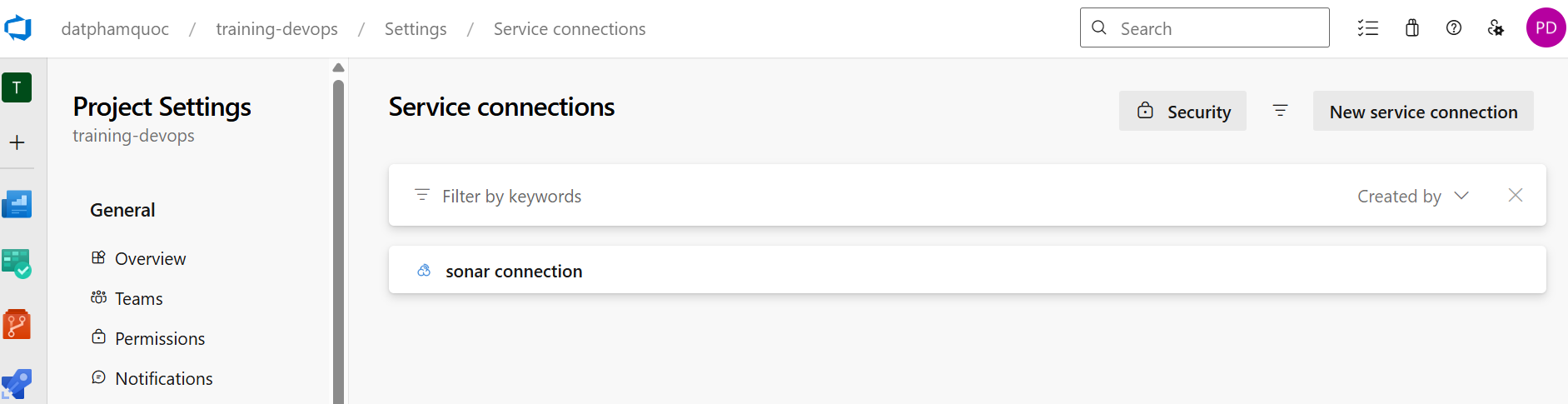


# Enter token name then click Generate Token



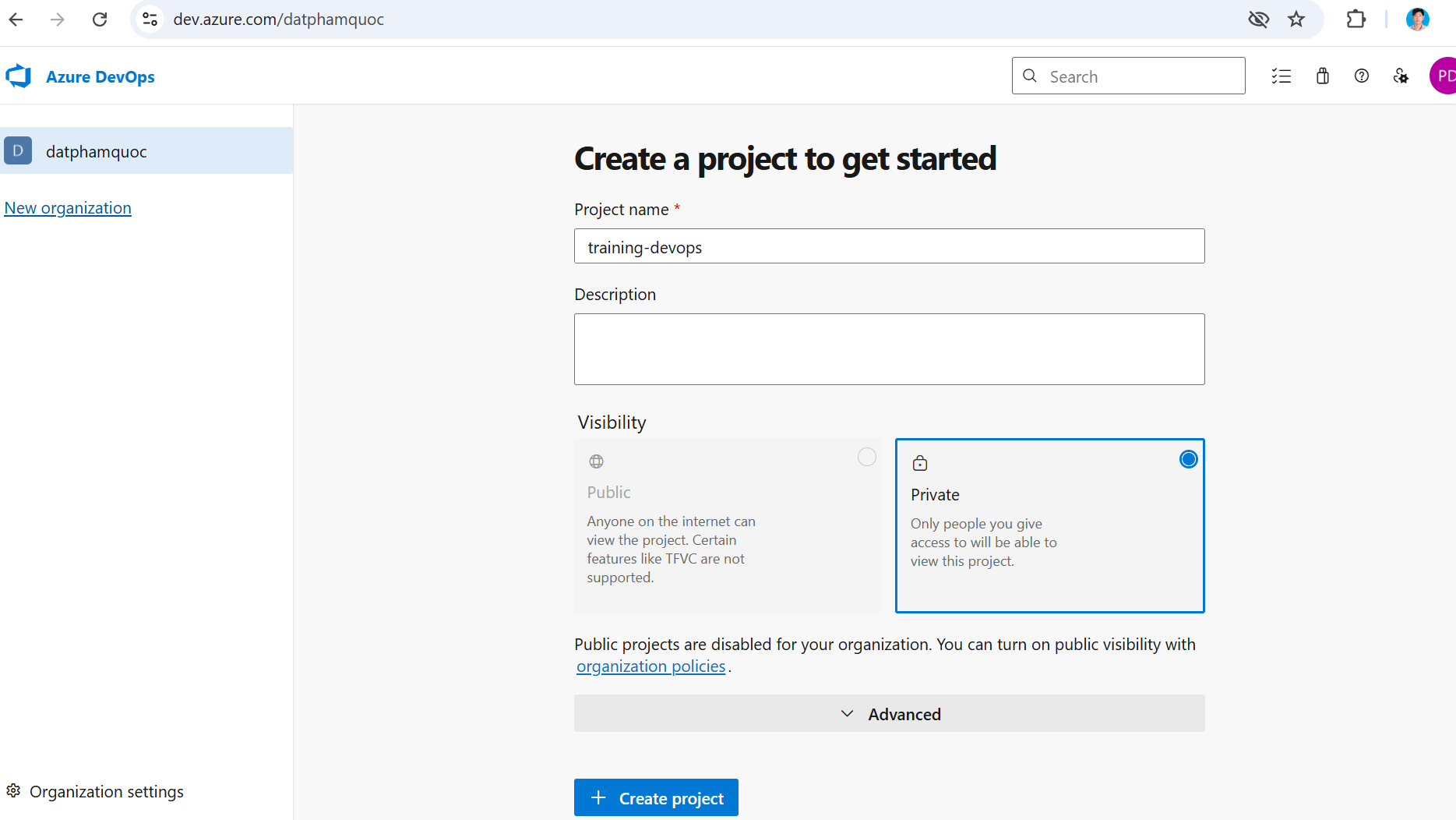


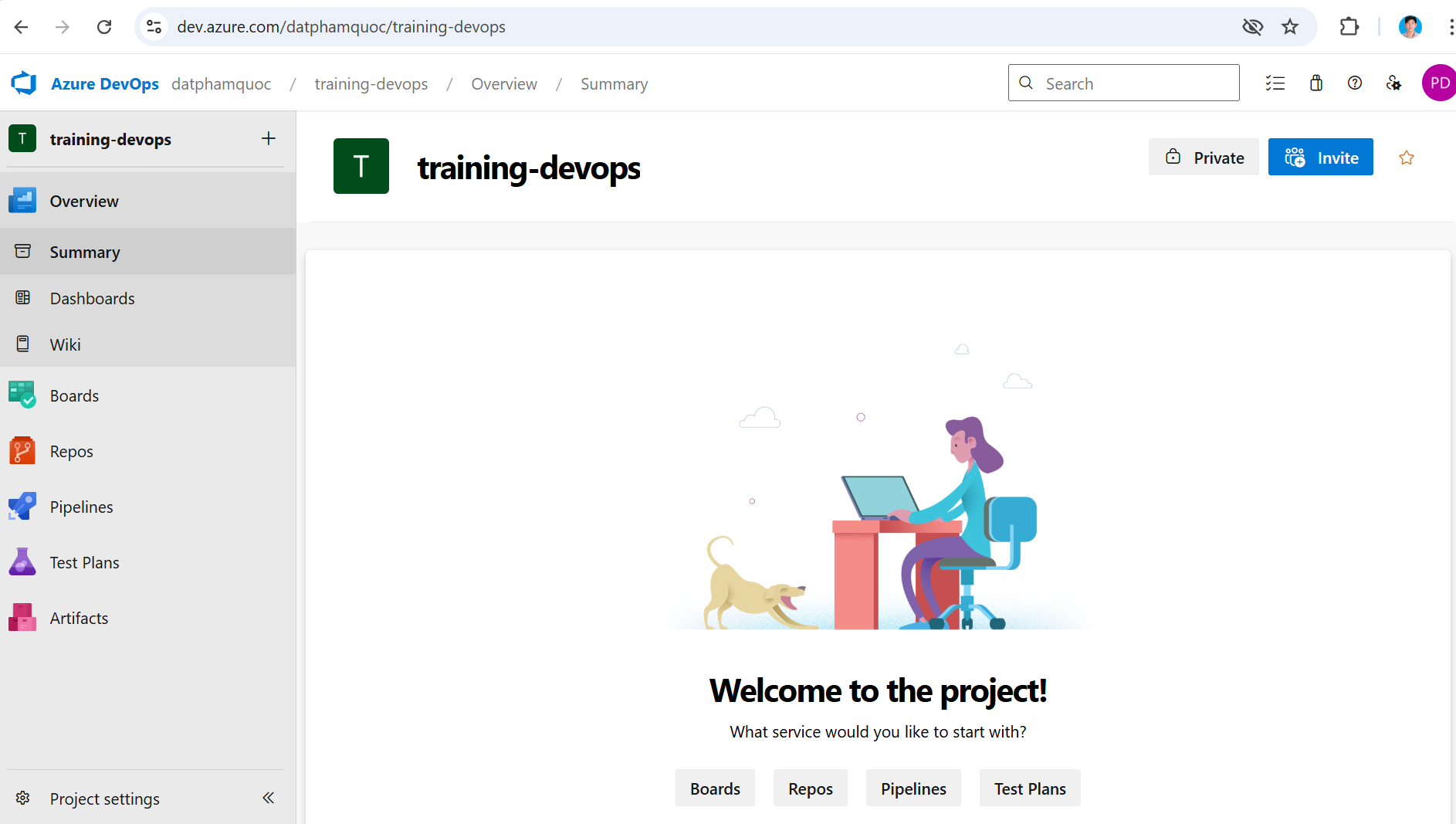
# Navigate to Azure devops page <https://dev.azure.com/datphamquoc/>



**Create project in Azure devops**

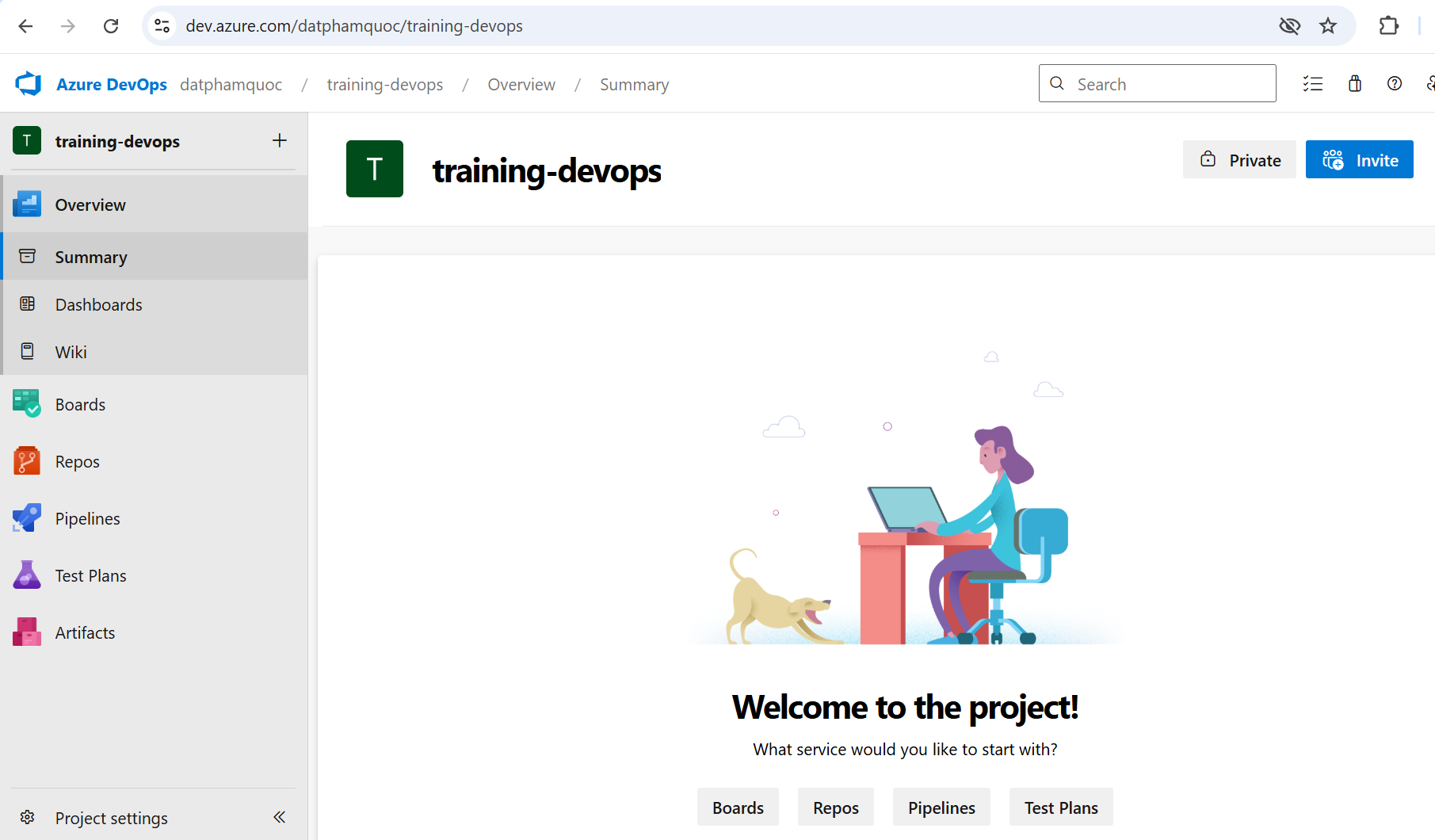
# Navigate to home page



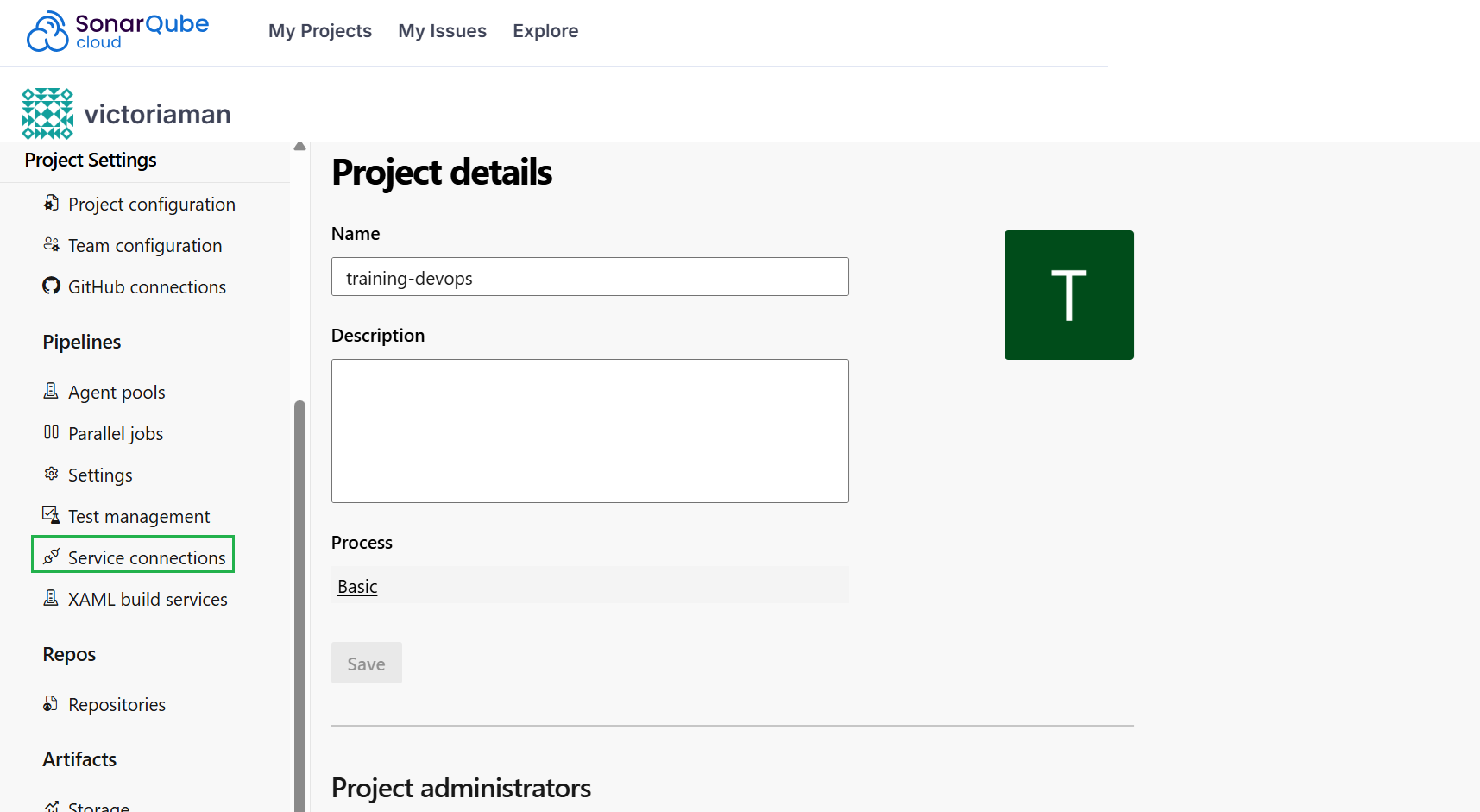


**Create Sonar cloud service connection**

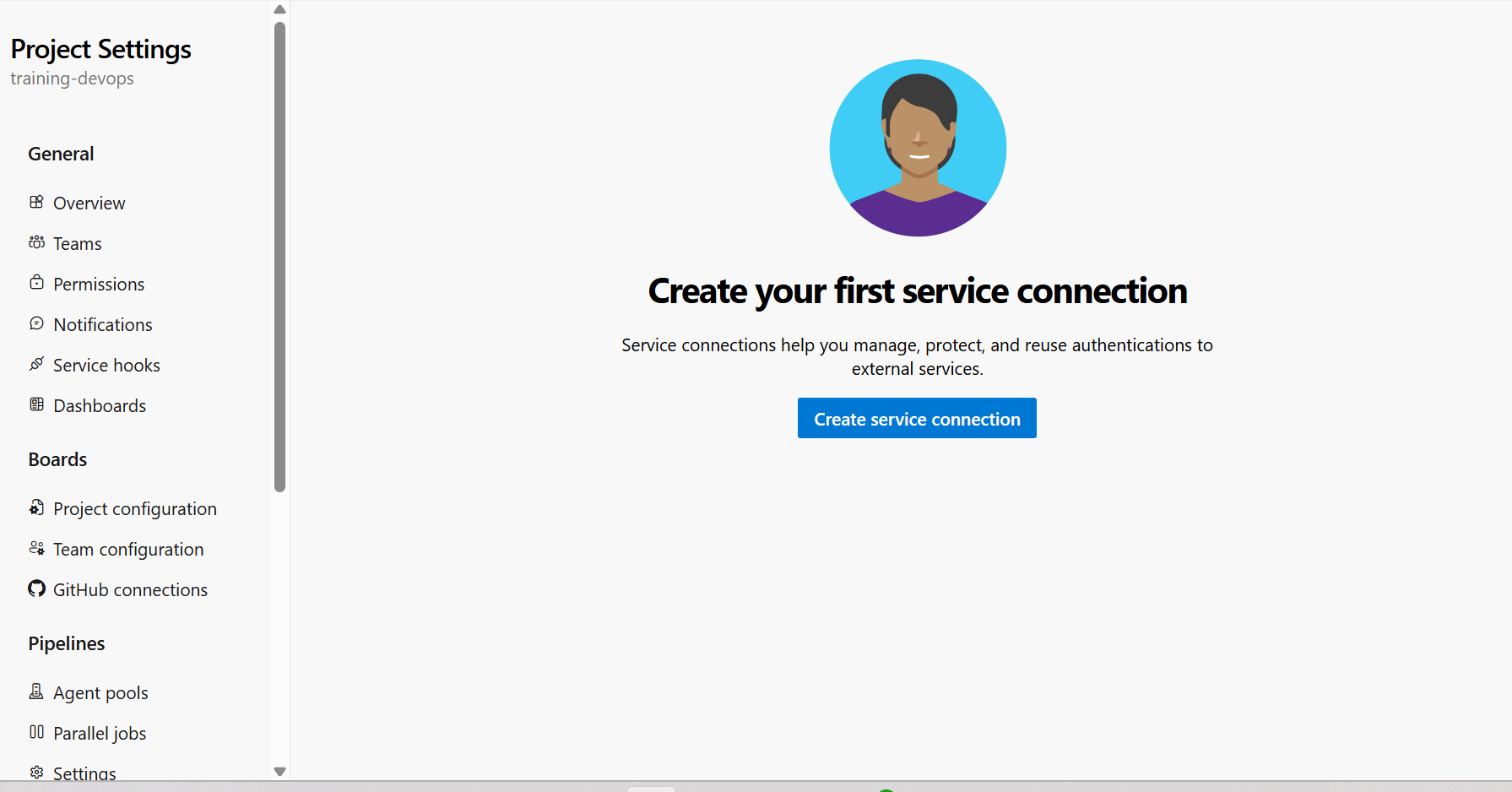
# In the training-devops project, click Project settings menu



# Click Service connections menu

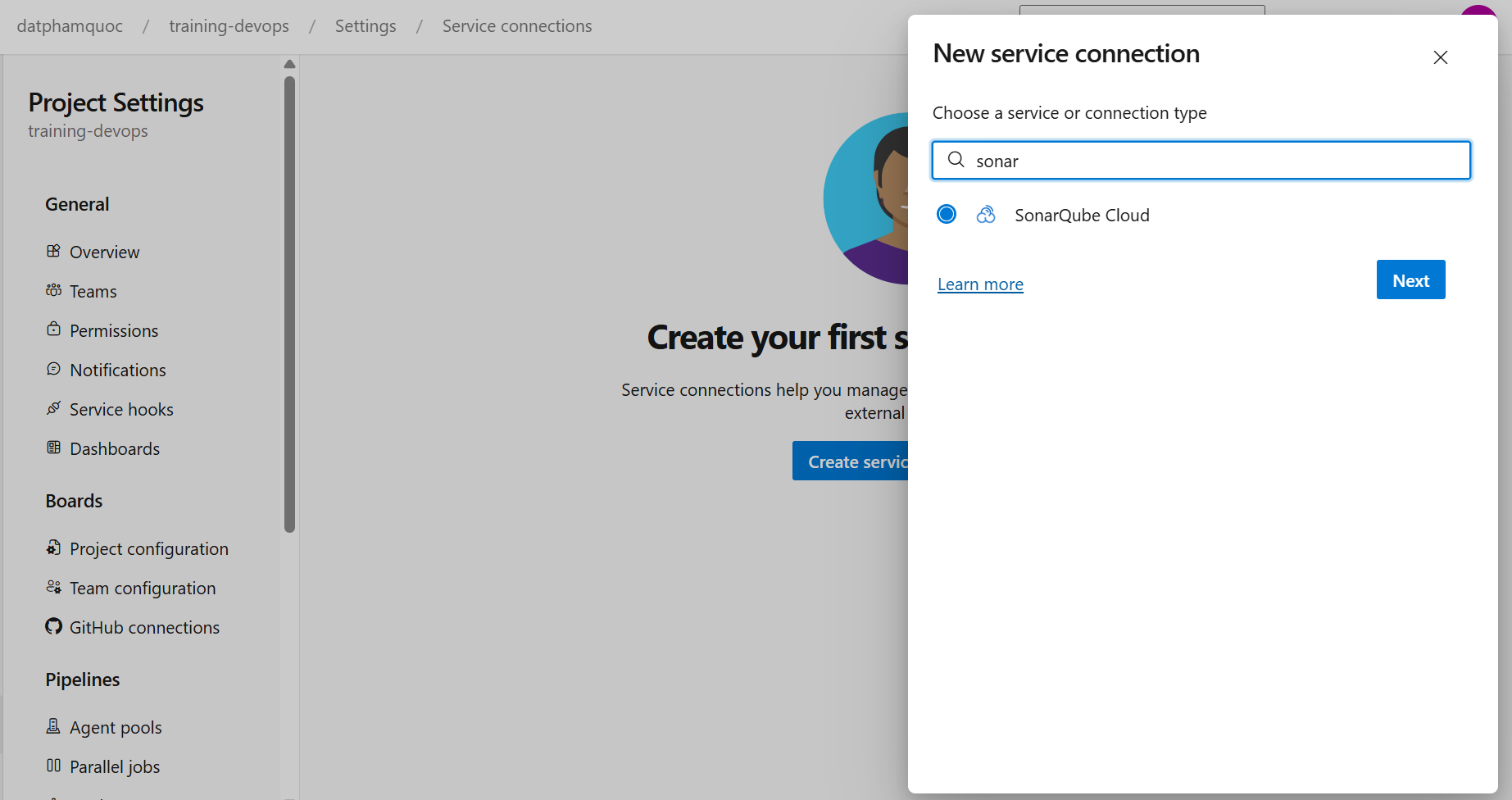


Click Create service connection

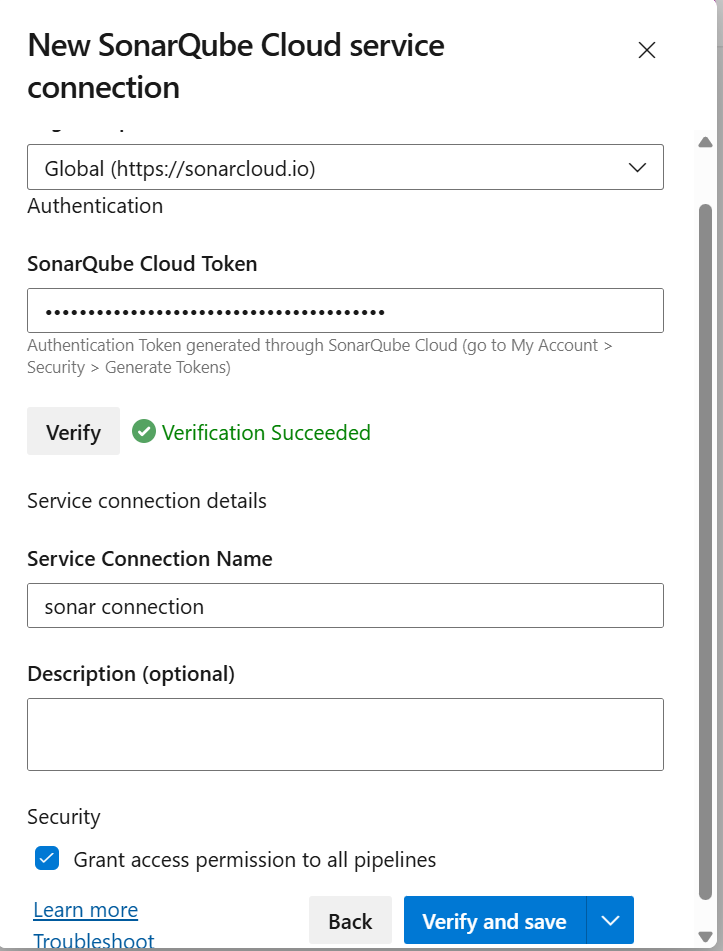


Note: Before create service connect to Sonar cloud, you **must install the SonarQube or SonarCloud extension** into Azure DevOps first

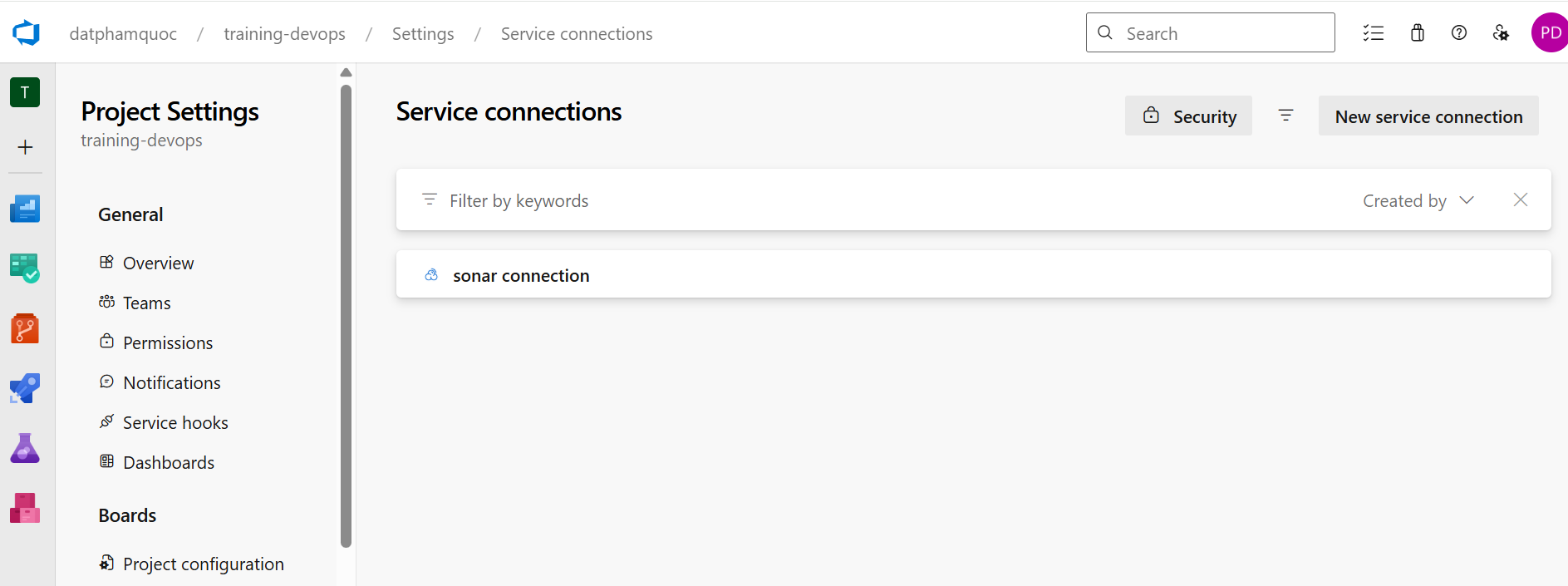


****

# Paste Sonar cloud token that is gotten from previous step then click Verify for verifying



New connection is created

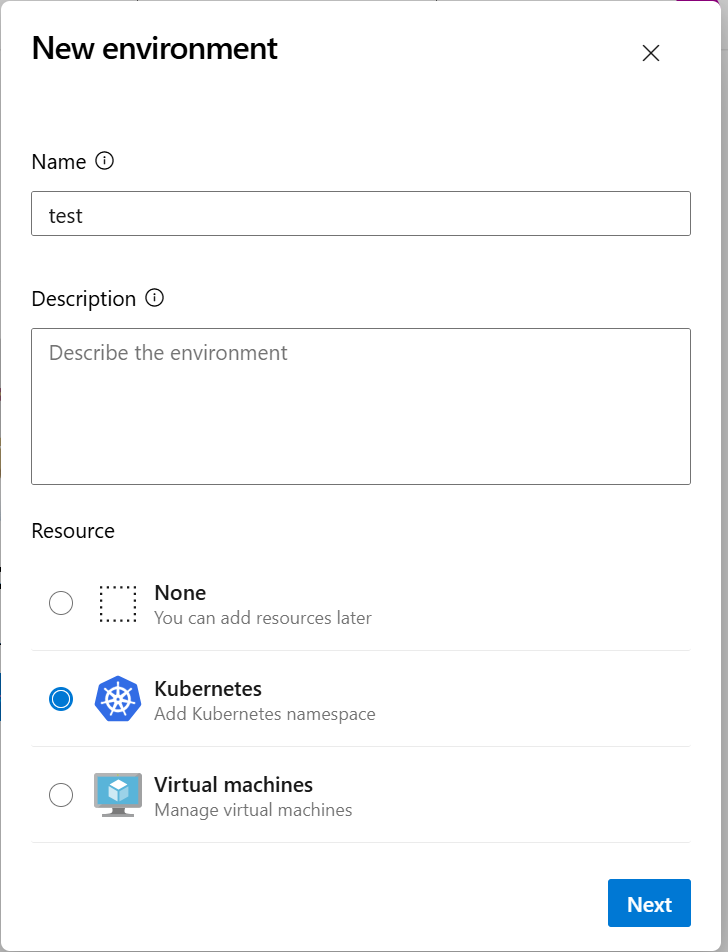


**Create Environment (This step only required when deploy app to AKS)**

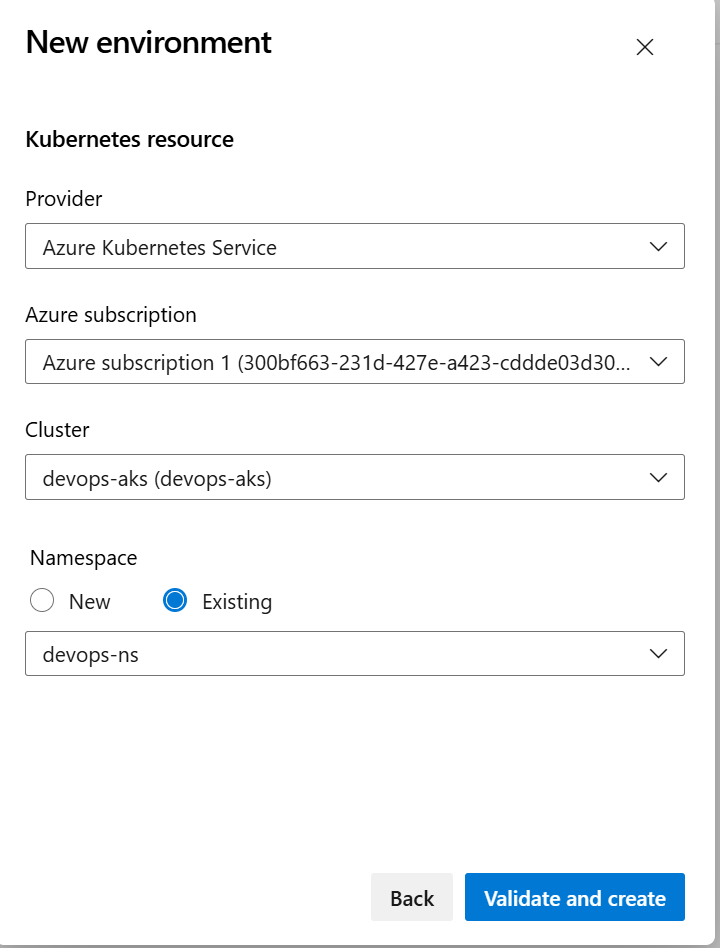
Go to Azure DevOps > Pipelines > Environments, click Create environment



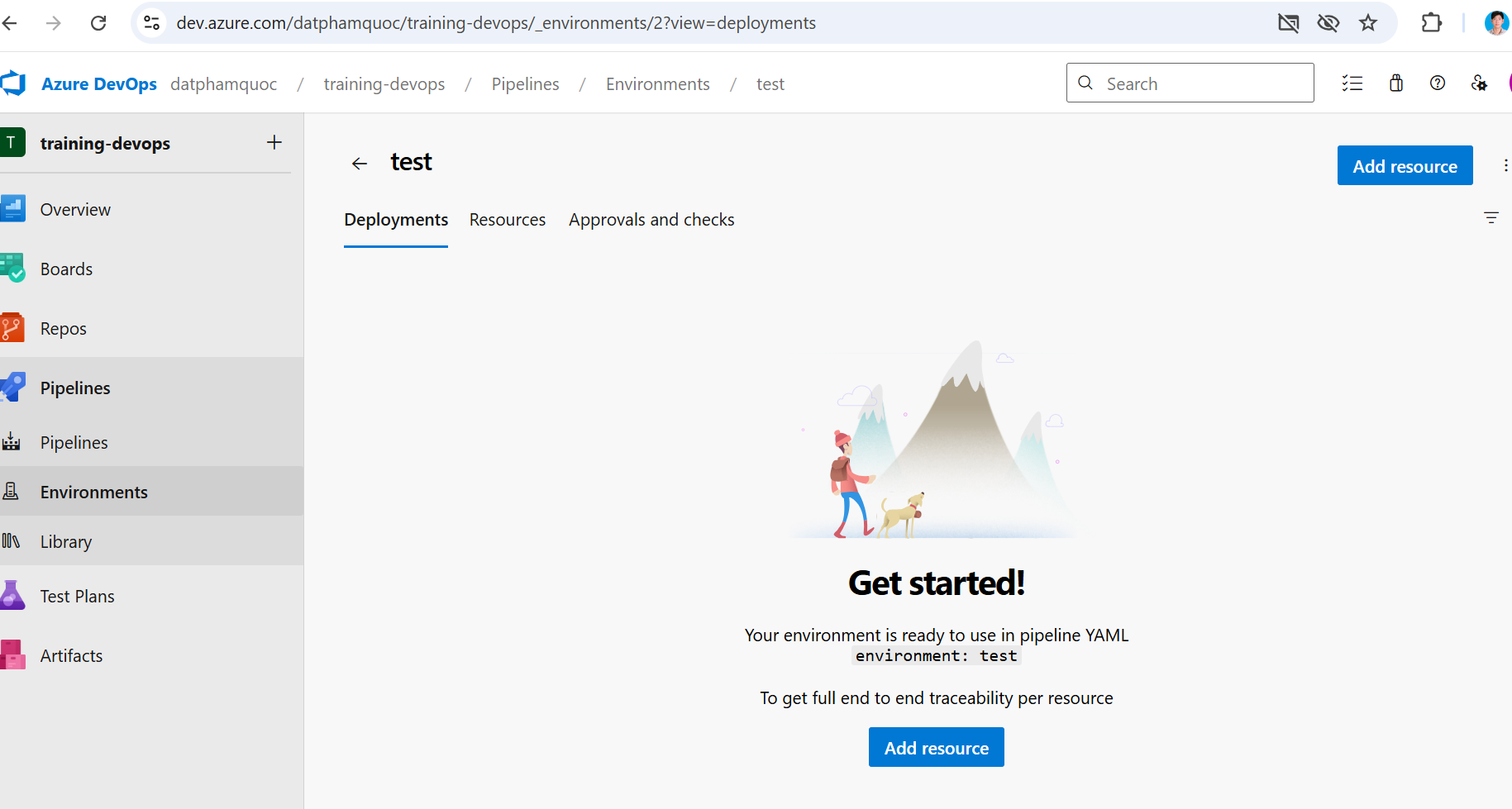
Input information as below, click Next



Choose information as below, click Validate and create

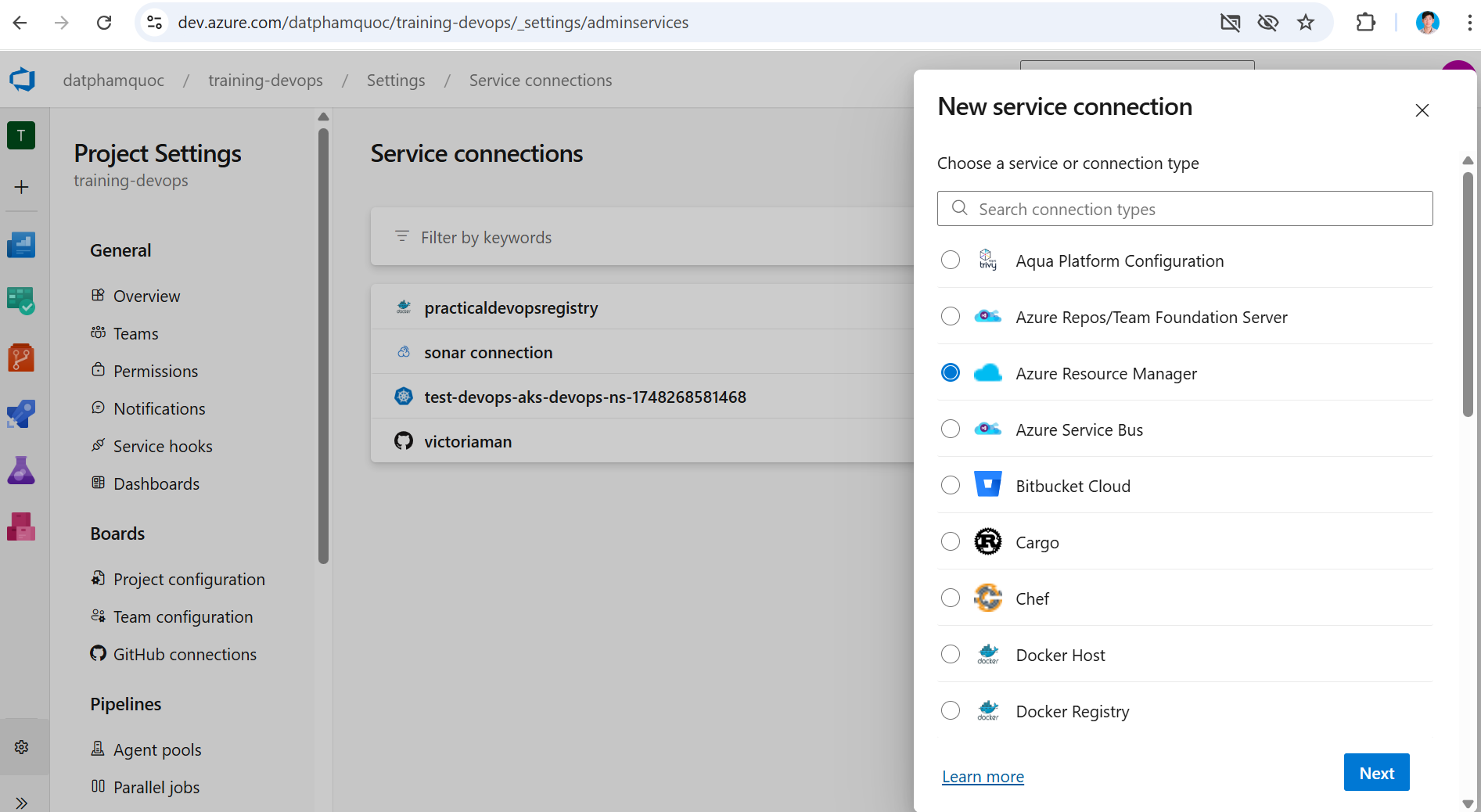


Test environment is created

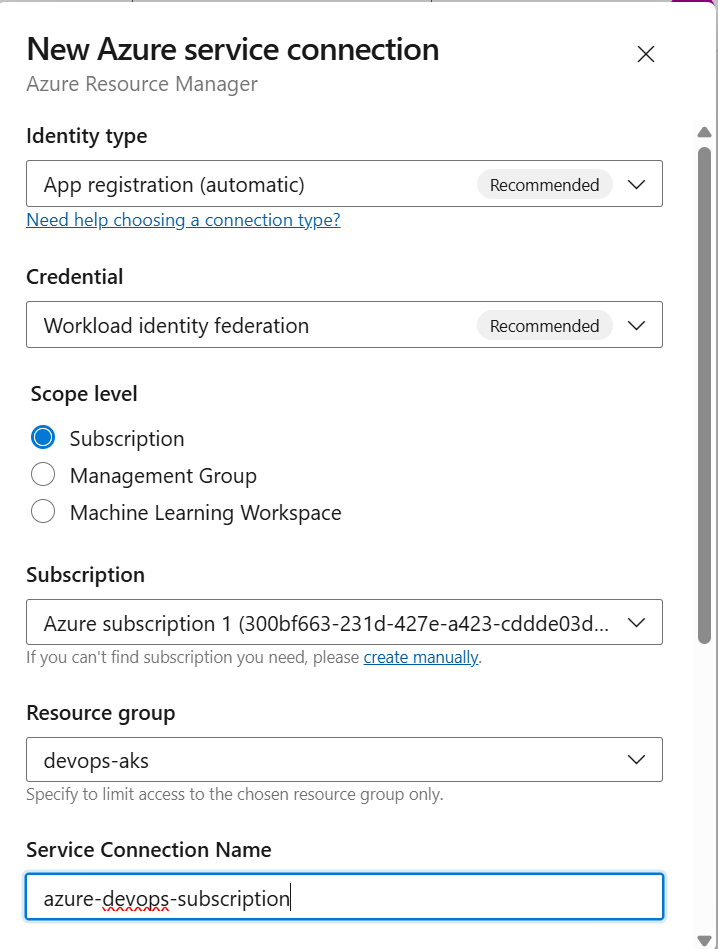


**Create Azure service connection (used for AzureCLI@2 task in pipeline)**

Choose Azure Resource Manager => Next



Input as below

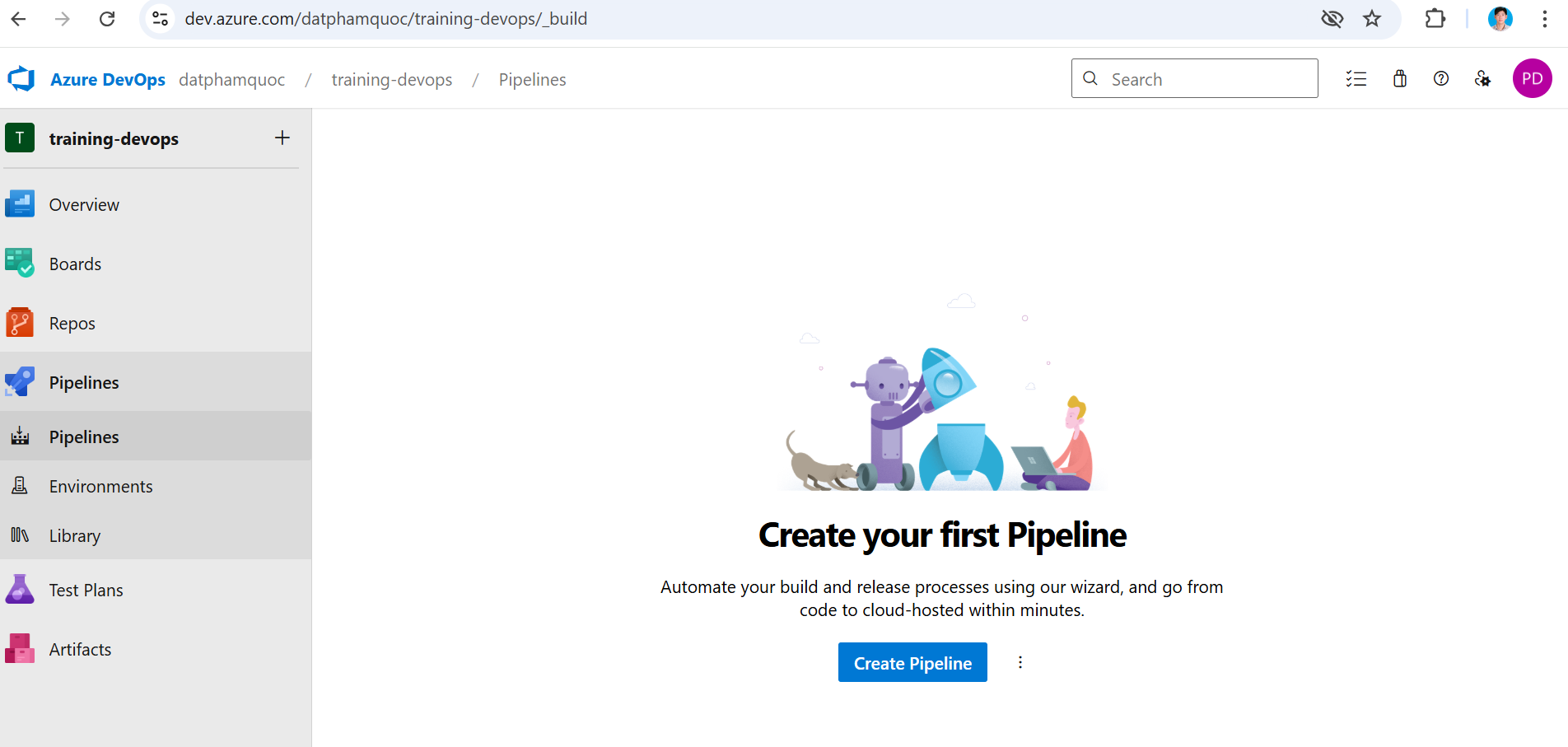


Click Save, new connection is created

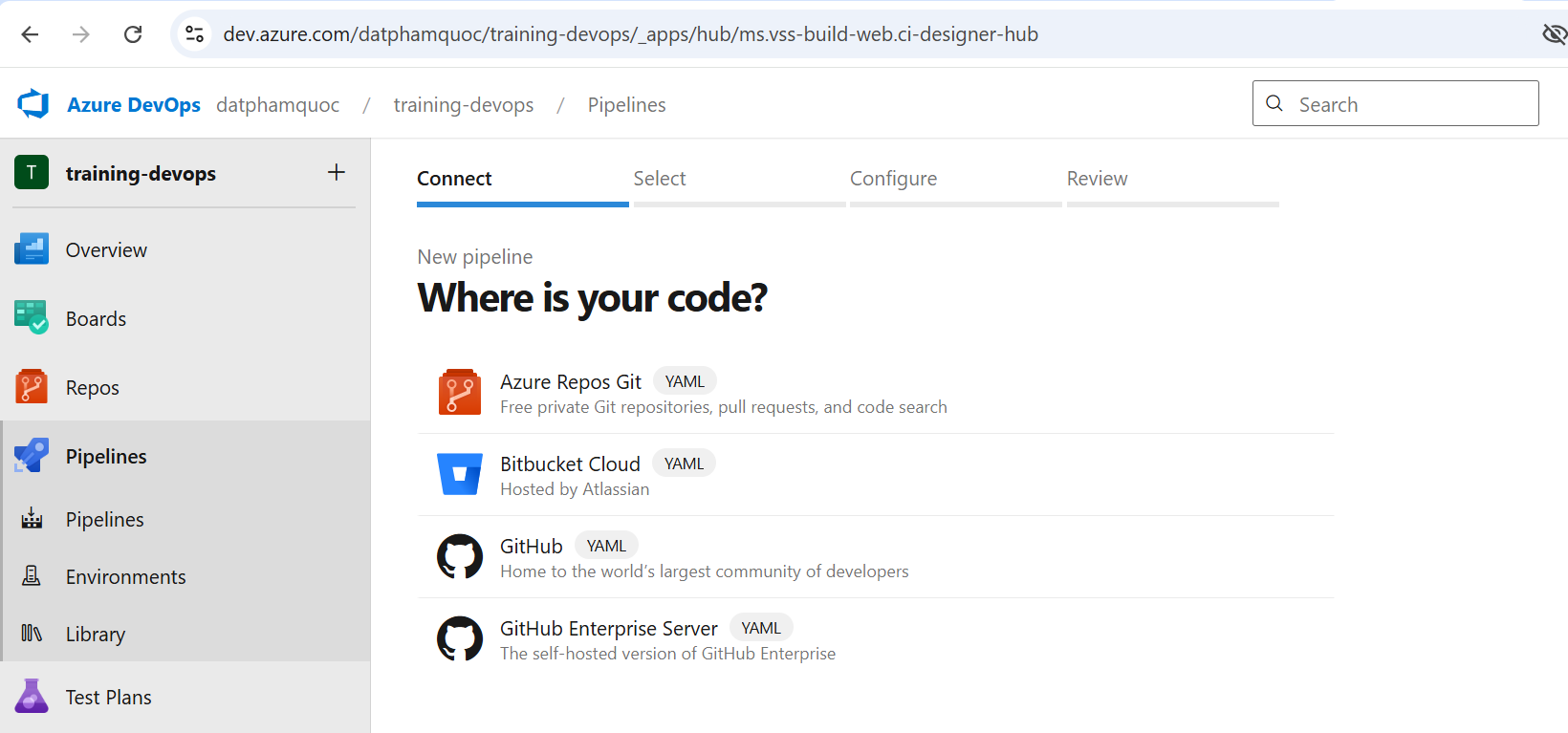


**Create pipeline**

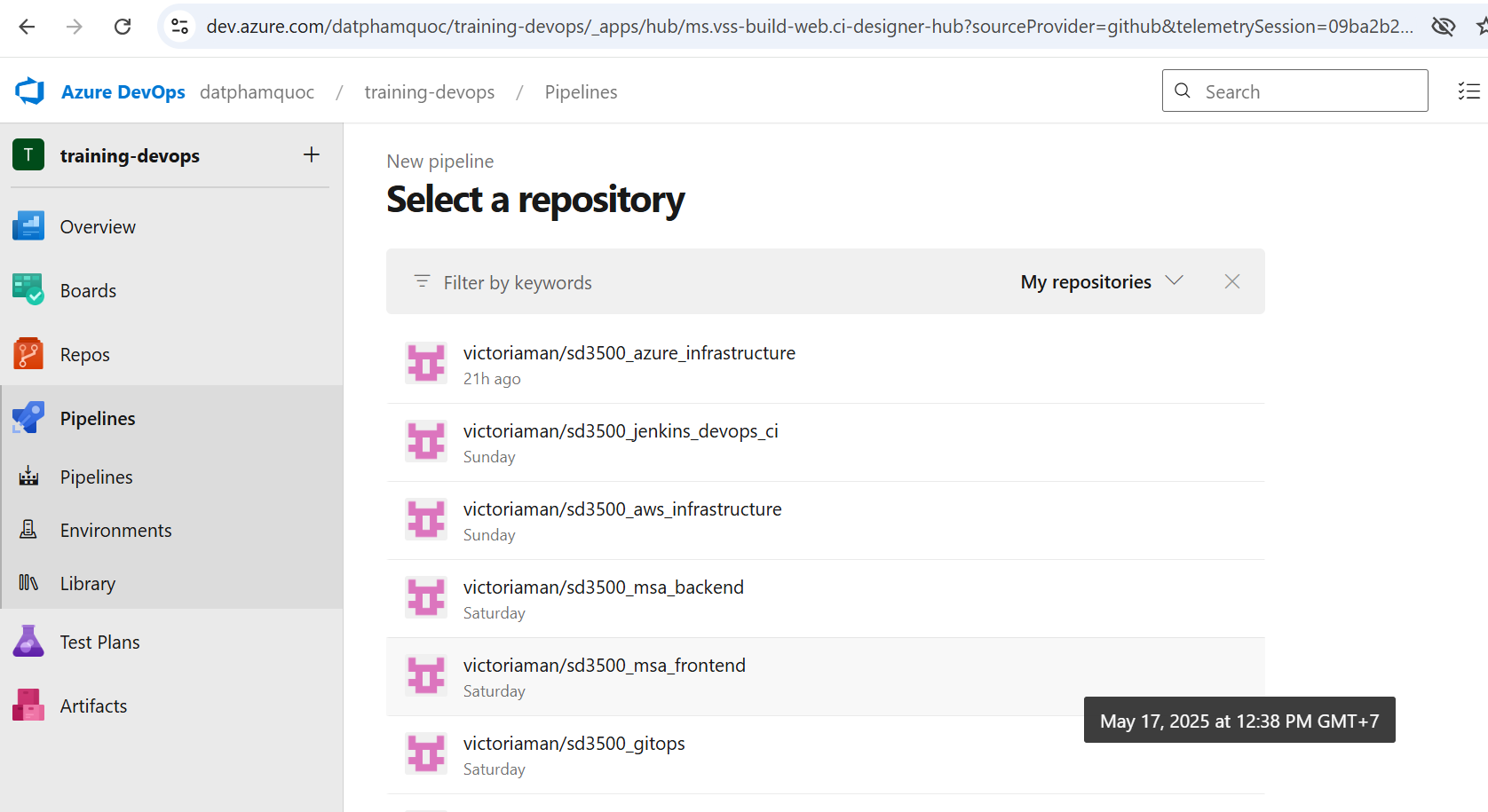
# Click Pipelines menu => Create Pipeline



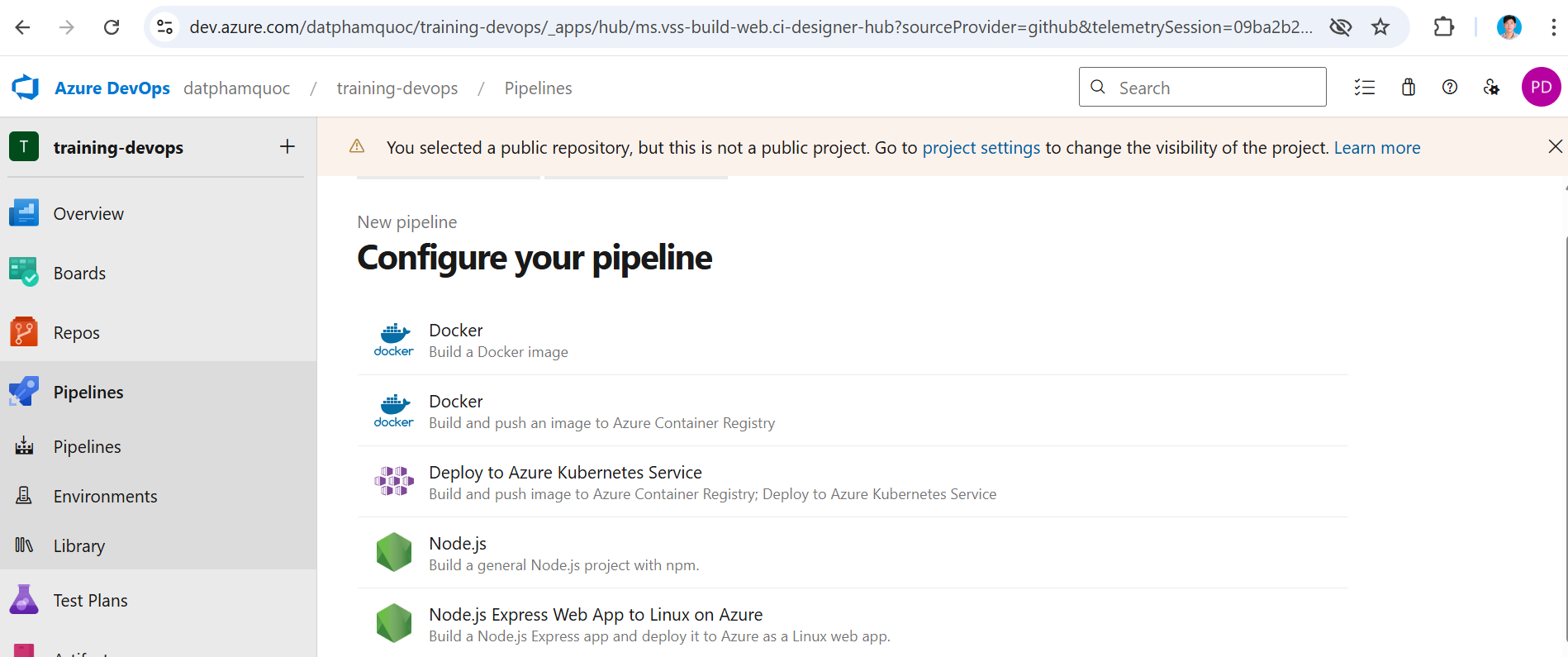
# Pick Github



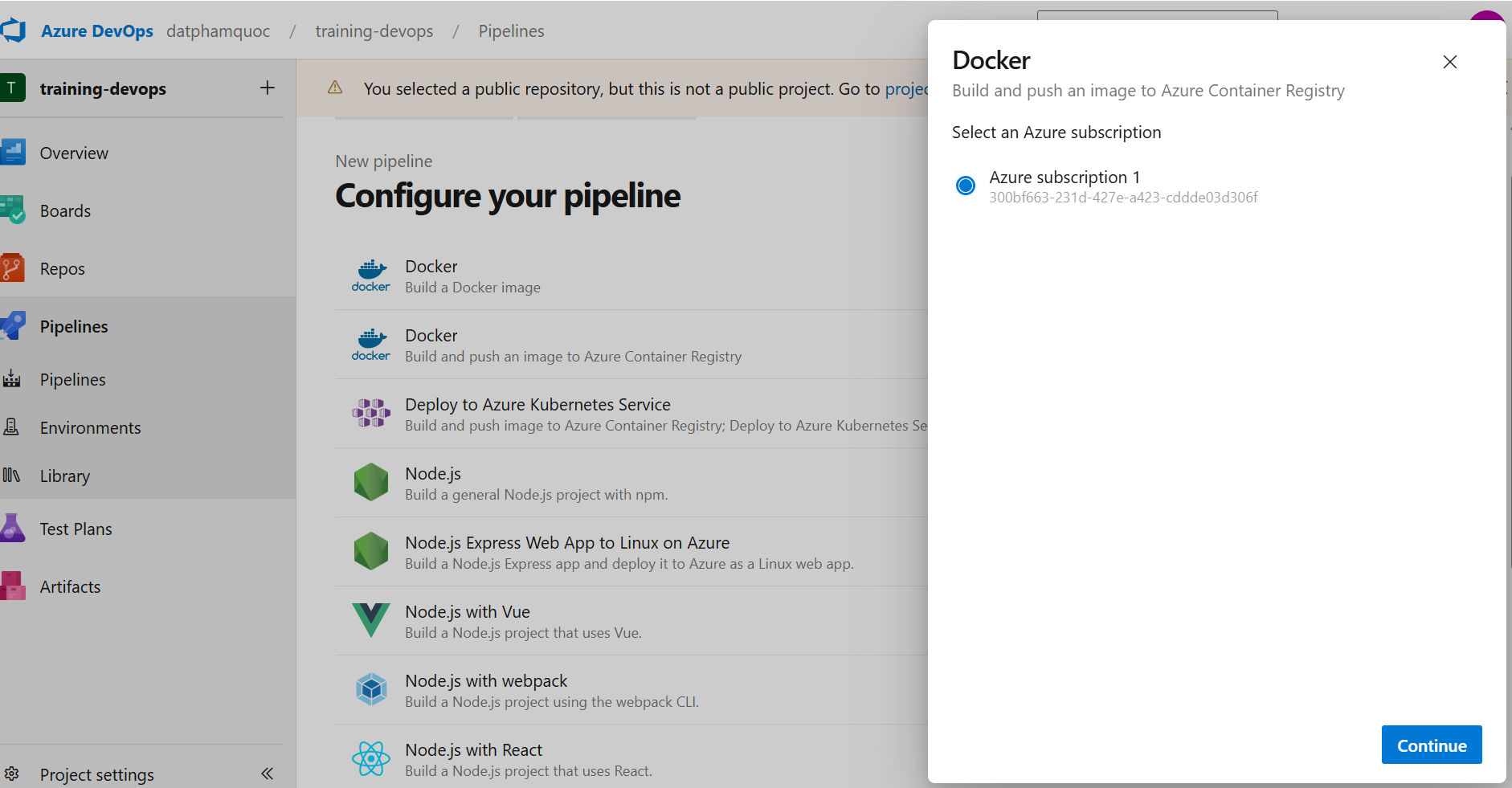
# Select sd3500\_msa\_frontend



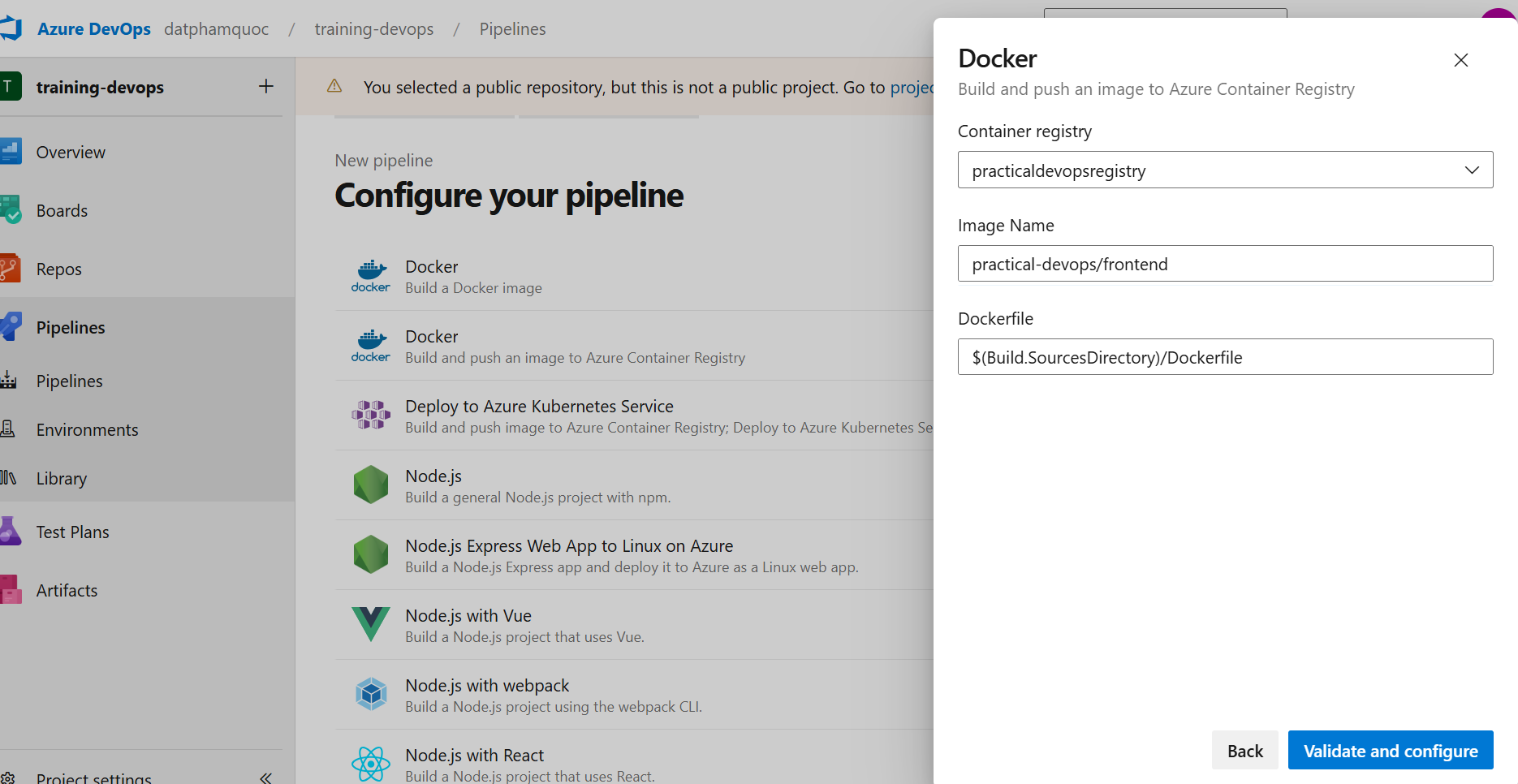
# Pick Docker (Build and push an image to Azure Container Registry)



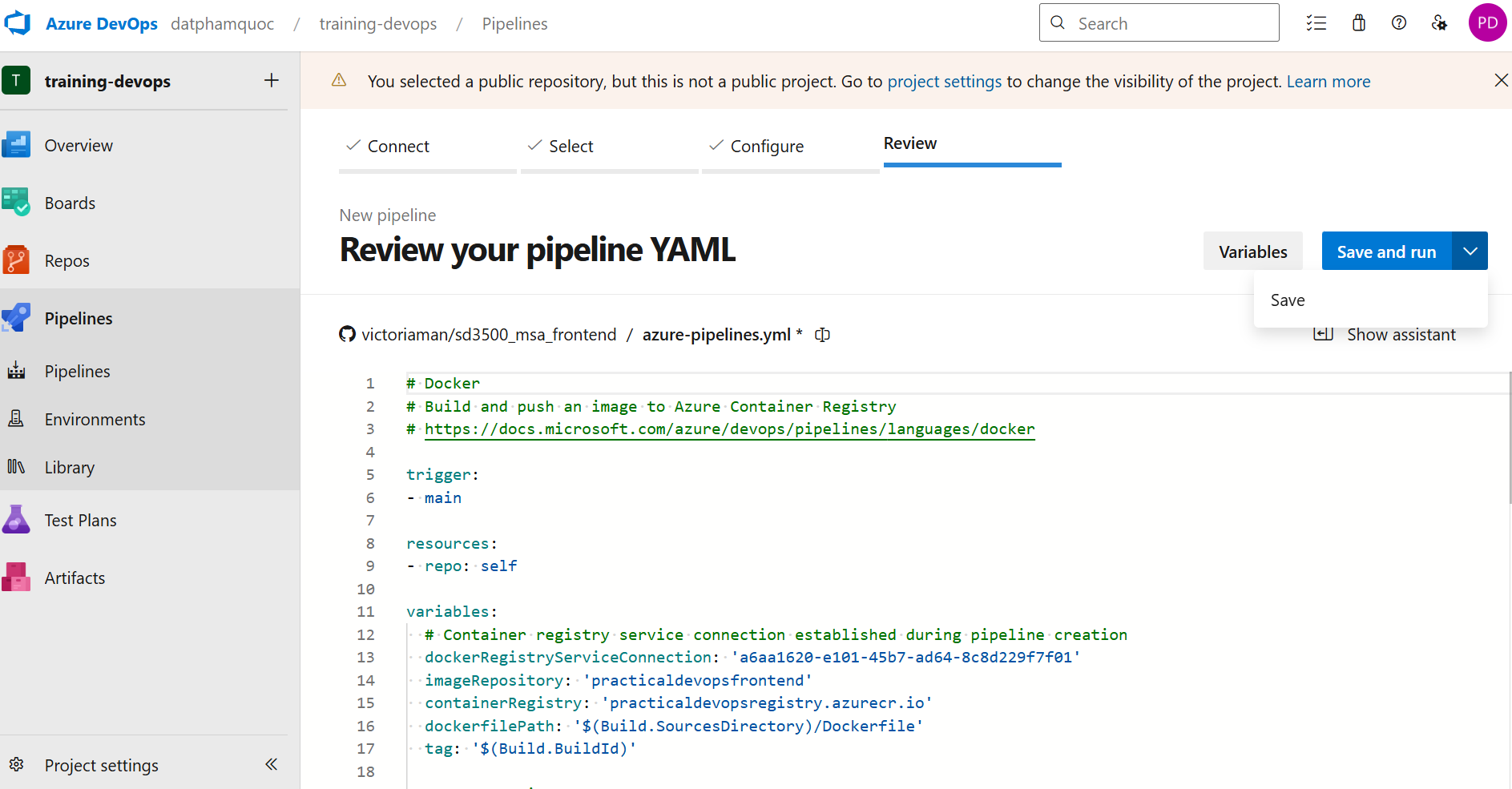
# Choose subscription -> Continue

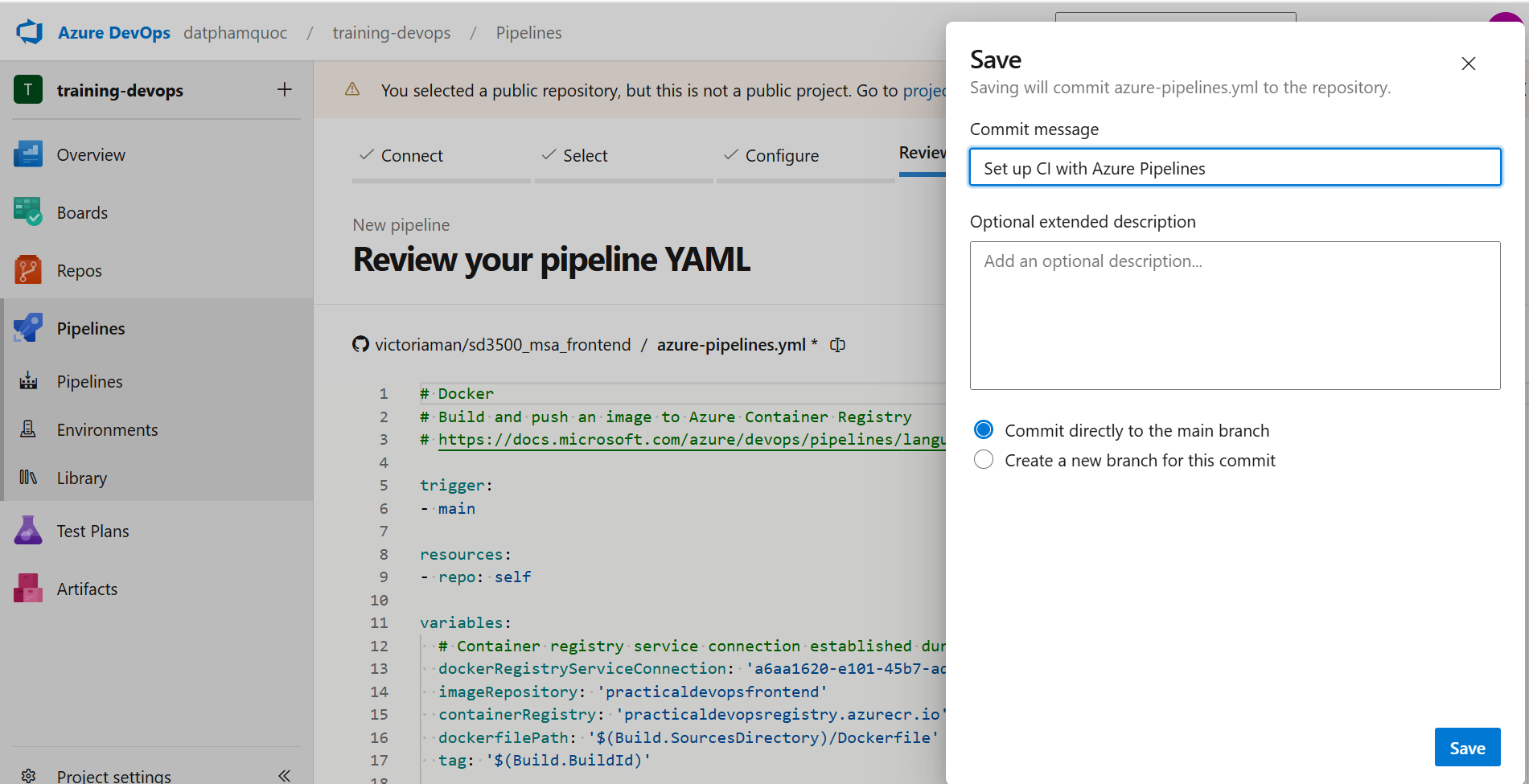


# Choose Container registry, image name and location of Dockerfile => Validate and configure



# Click Save to create azure-pipeline.yml (we will adjust this pipeline to add another step as trivy, Sonar scan)





azure-pipelines.yml is generated in Github

