

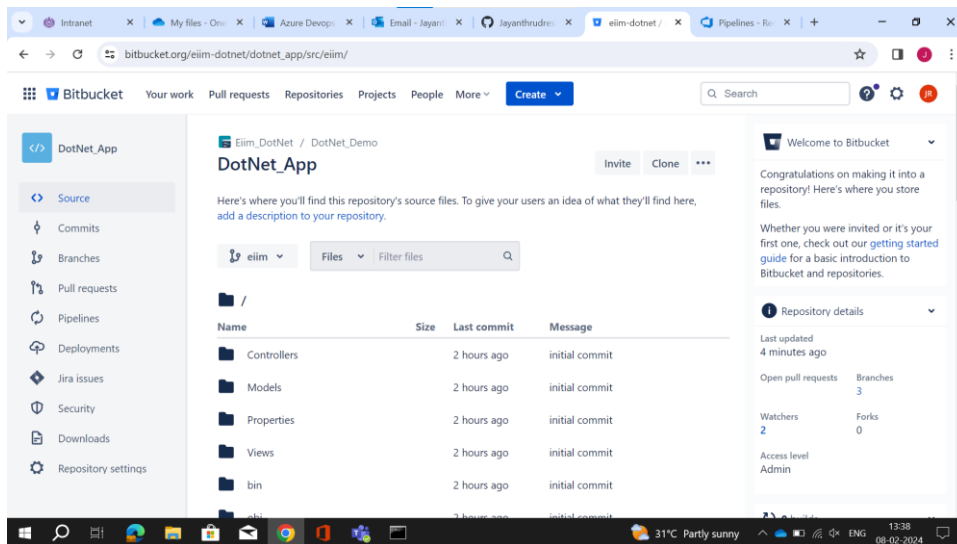
# Azure Devops Yaml

Jayanth R

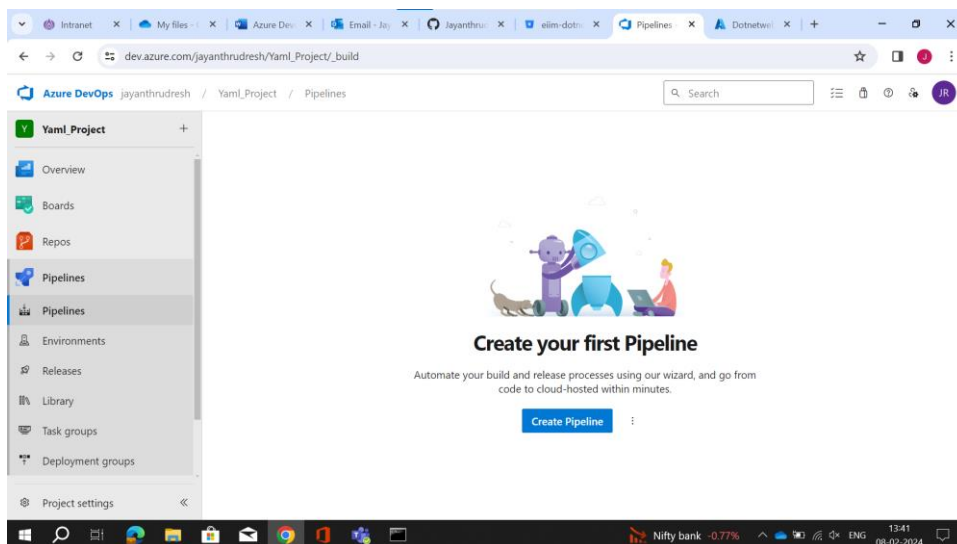
ACE11696

Here are the steps for Azure Devops CI CD Using Yaml.

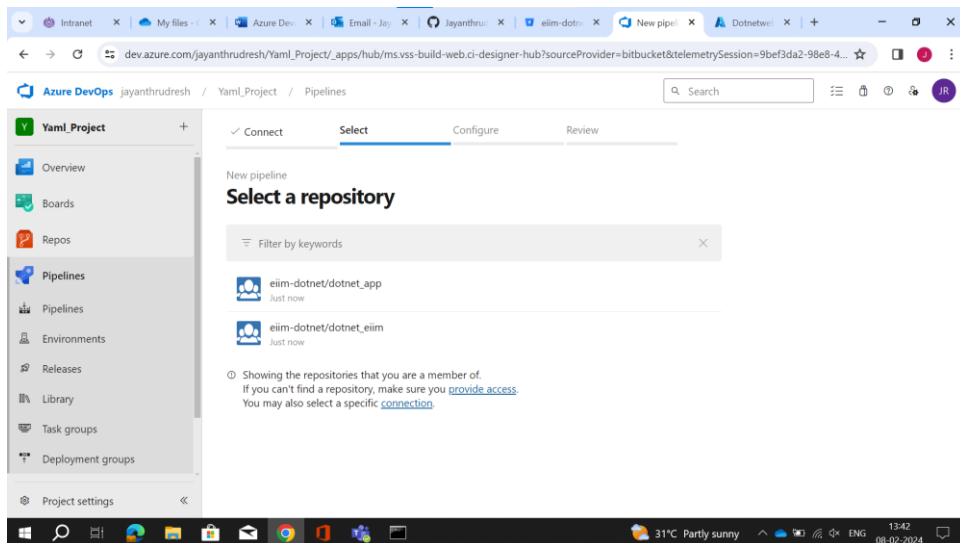
Step 1: This is my BitBucket Repository containing my dotnet application.



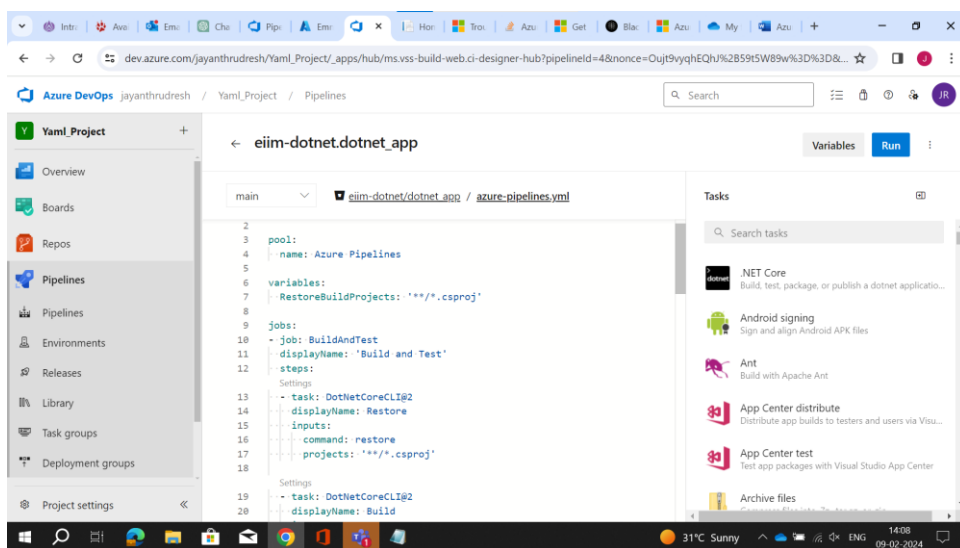
Step 2: I'm creating a pipeline for accessing my repository to build and deploy it.



Step 3: Select Your Private repository as your project is private in Azure DevOps.



Step 4: Now Create your own Yml file to add 2 jobs for build and deploy it, this is for build.



Step 5: This is the deploy job in Yml.

The screenshot shows the Azure DevOps web interface for a pipeline named 'eiim-dotnet.dotnet\_app'. The left sidebar contains a navigation menu with options like Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, and Deployment groups. The main area displays the pipeline's YAML configuration for the 'main' branch. The configuration includes a 'Deploy' job that depends on 'BuildAndTest'. It contains two tasks: 'DownloadPipelineArtifact@2' and 'AzureRmWebAppDeployment@4'. The 'AzureRmWebAppDeployment@4' task is configured with inputs for 'ConnectionType', 'azureSubscription', 'appName', 'webAppName', and 'packageForLinux'. A 'Tasks' panel on the right lists various available tasks such as '.NET Core', 'Android signing', 'Ant', 'App Center distribute', 'App Center test', and 'Archive files'. The bottom status bar shows the system time as 14:09 on 09-02-2024.

```
46 - condition: succeededOrFailed()
47
48 - job: Deploy
49   displayName: 'Deploy'
50   dependsOn: BuildAndTest
51   steps:
52     - task: DownloadPipelineArtifact@2
53       inputs:
54         buildType: 'current'
55         artifactName: 'Eiim-DotNet'
56         targetPath: '$(System.DefaultWorkingDirectory)'
57     - task: AzureRmWebAppDeployment@4
58       inputs:
59         ConnectionType: 'AzureRM'
60         azureSubscription: 'Free Trial(4da72062-251b-49f1-8a37-3d969715a571)'
61         appType: 'webAppLinux'
62         webAppName: 'EimYamlApp'
63         packageForLinux: '$(System.DefaultWorkingDirectory)/**/*.zip'
```

Step 6: The Pipeline is ready and start the run here.

The screenshot shows the 'Pipelines' view in the Azure DevOps interface. The left sidebar is the same as in the previous image. The main area is titled 'Pipelines' and includes a 'New pipeline' button and a 'Filter pipelines' dropdown. Below this is a section titled 'Recently run pipelines' which contains a table. The table has two columns: 'Pipeline' and 'Last run'. It lists the 'eiim-dotnet.dotnet\_app' pipeline, which has 'No runs yet'. The bottom status bar shows the system time as 14:57 on 08-02-2024.

Pipeline	Last run
eiim-dotnet.dotnet_app	No runs yet

Step 7: The yml file code is given as 2nd job should run if 1st job runs only.

The screenshot shows the Azure DevOps web interface for a pipeline named 'Update azure-pipelines.yml for Azure Pipelines'. The pipeline is triggered manually by 'Jayanth Rudresh'. The summary shows the repository as 'eim-dotnet-dotnet\_app' and the build ID as '20240208.1'. The jobs table shows two jobs: 'Build and Test' and 'Deploy', both with a status of 'Queued'.

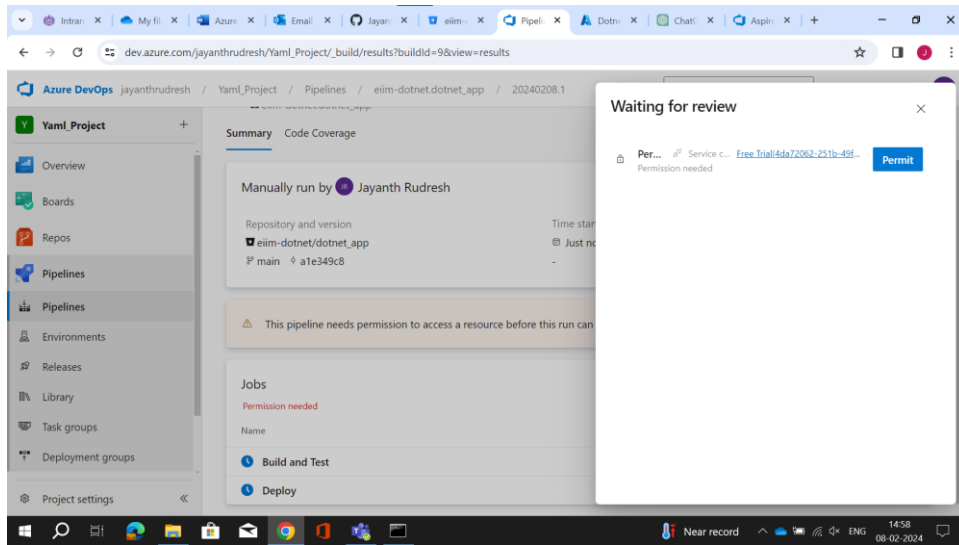
Name	Status	Duration
Build and Test	Queued	
Deploy	Queued	

Step 8: Now i have have to give access for a resource before running.

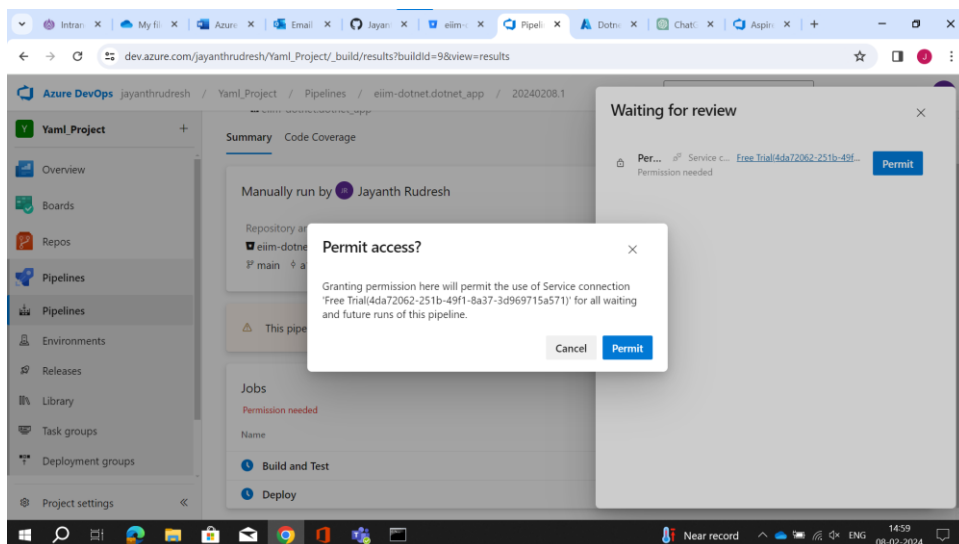
The screenshot shows the same pipeline run, but now it has failed due to a permission issue. A message states: 'This pipeline needs permission to access a resource before this run can continue'. The jobs table shows the status of the jobs has changed to 'Waiting'.

Name	Status	Duration
Build and Test	Waiting	
Deploy	Waiting	

## Step 9: Permit Access.



## Step 10: Permit.



Step 11: Now you can see here 1<sup>st</sup> job failed so 2<sup>nd</sup> job is stopped running.

The screenshot shows the Azure DevOps interface for a pipeline named 'eiim-dotnet.dotnet\_app'. The 'Build and Test' job failed with the error 'No files matched the search pattern. Build and Test - Restore'. The 'Deploy' job is marked as 'Skipped'.

Name	Status	Duration
Build and Test	Failed	12s
Deploy	Skipped	

Step 12: Here my 2 jobs also run successfully, and my application is successfully deployed

The screenshot shows the Azure DevOps interface for a pipeline named 'eiim-dotnet.dotnet\_app'. Both the 'Build and Test' and 'Deploy' jobs completed successfully. The 'Deploy' job is expanded, showing the 'AzureRmWebAppDeployment' task details.

Name	Status	Duration
Build and Test	Completed	32s
Deploy	Completed	1m 49s

**AzureRmWebAppDeployment**

```
1 Starting: AzureRmWebAppDeployment
2 =====
3 Task : Azure App Service deploy
4 Description : Deploy to Azure App Service a web, mobile, or API app using Docker, Java, .NET, .NET Core
5 Version : 4.234.2
6 Author : Microsoft Corporation
7 Help : https://aka.ms/azureappsservicetroubleshooting
8 =====
9 Get service connection details for Azure App Service: 'emmiYamlApp'
10 Package deployment using ZIP deploy initiated.
11 Deploy logs can be viewed at https://emmiYamlApp.scm.azurewebsites.net/api/deployments/c7b6b2d1-8077-4...
12 Successfully deployed web package to App Service.
13 Successfully updated App Service configuration details
14 Successfully updated deployment History at https://emmiYamlApp.scm.azurewebsites.net/api/deployments/...
15 App Service Application URL: https://emmiYamlApp.azurewebsites.net
16 Finishing: AzureRmWebAppDeployment
```

Step 13: Now after Navigating to that link We can see our web app running.

The screenshot shows the web application running on the URL 'emmiYamlApp.azurewebsites.net'. The application displays a welcome message and a link to the Bitbucket code repository.

dotnet\_eiim

- [Home](#)
- [Privacy](#)

Welcome Aspire

Bitbucket code

© 2024 - dotnet\_eiim - [Privacy](#)