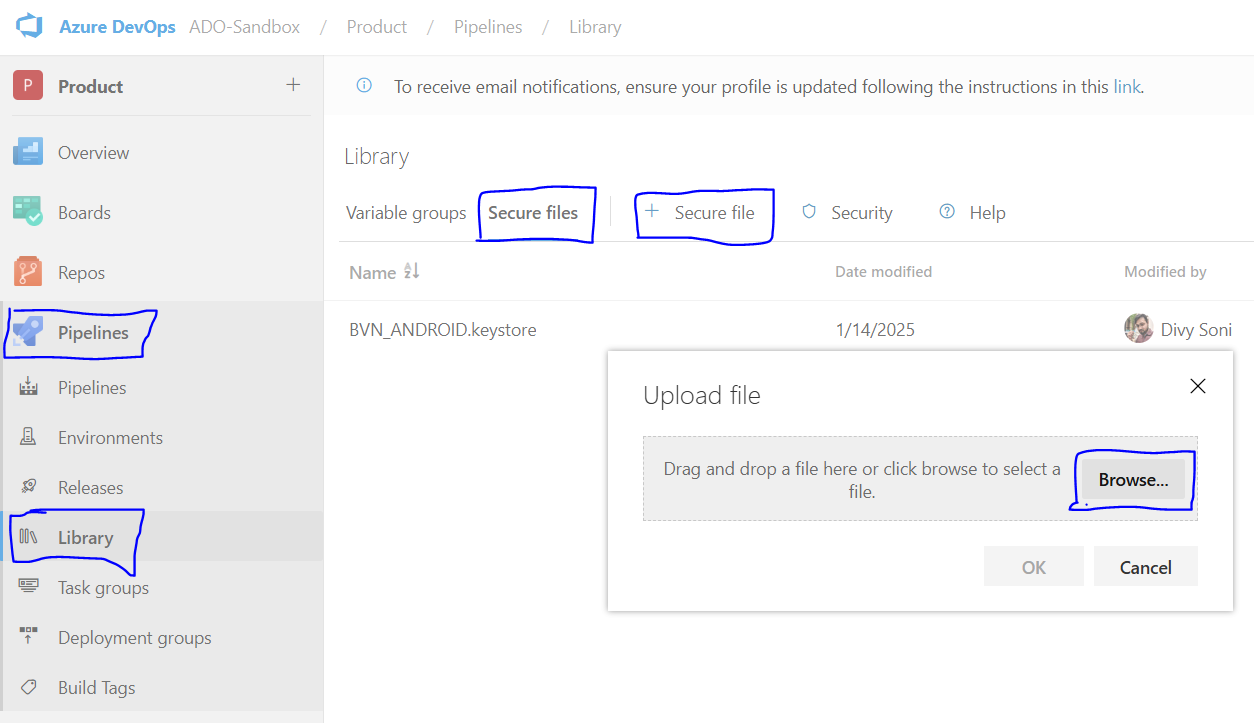
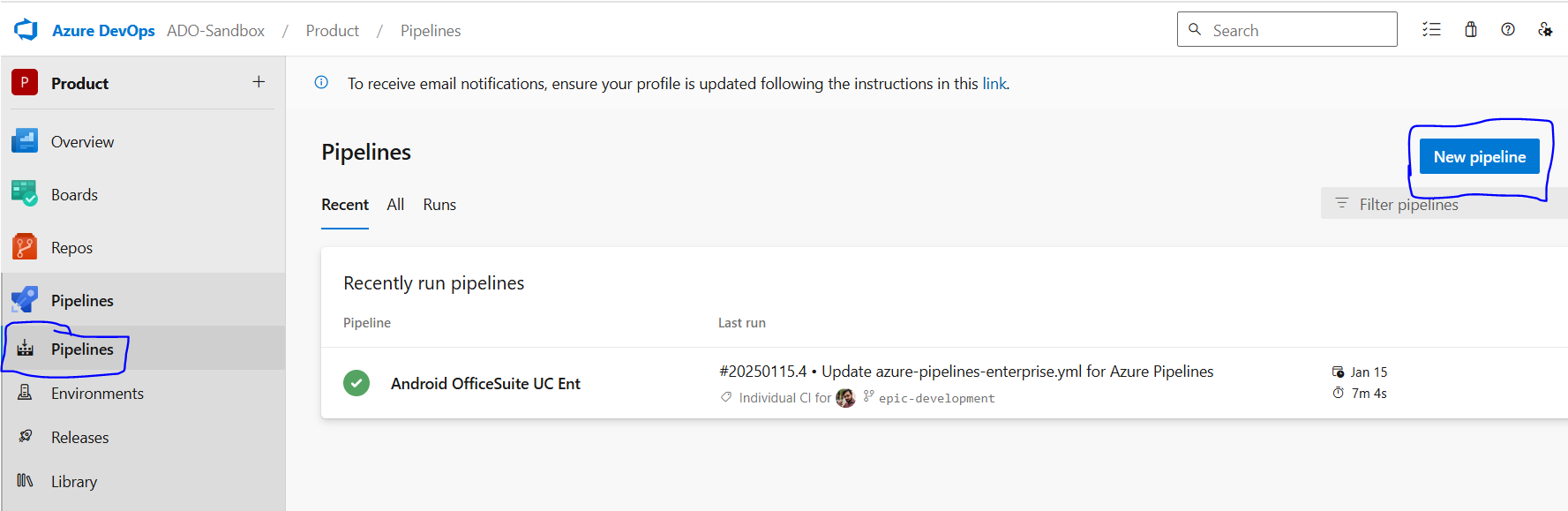
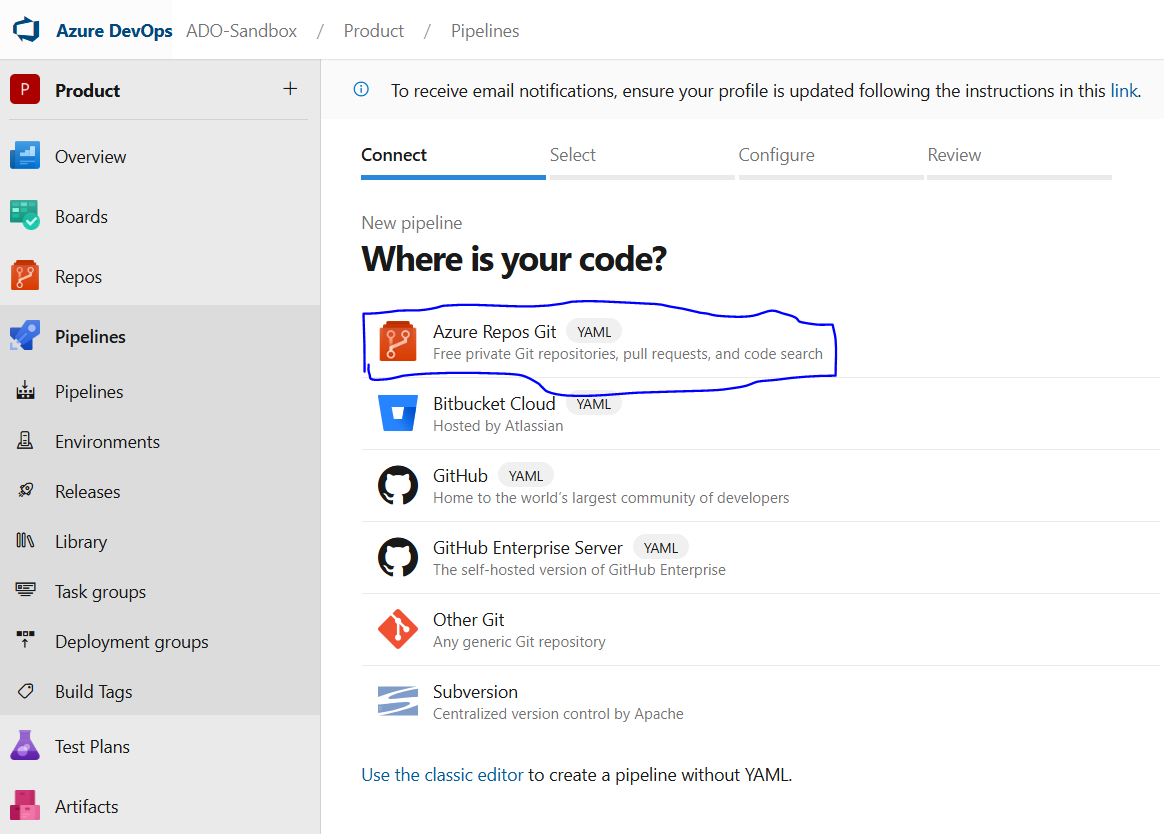
Creating APK

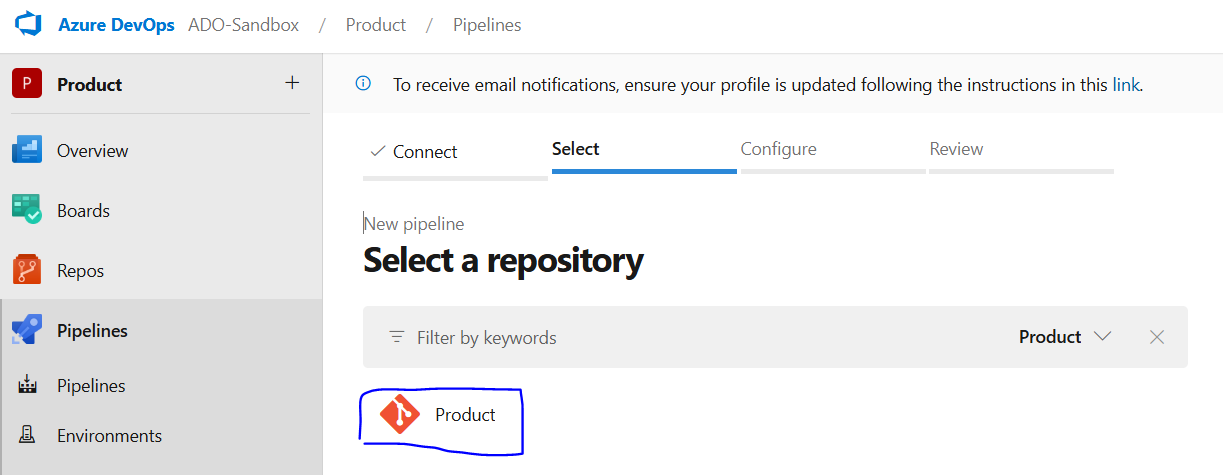
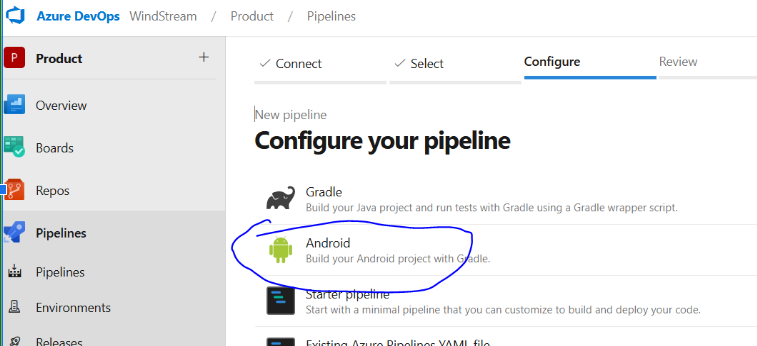
AzureDevOps Pipeline Setup for Android app:

1. Open Azure Devops
2. Click “Pipelines” -> “Library” -> “Secure files” -> “Secure file” -> “Browse”  
   Select Keystore file (in our code, the file name is ***BVN\_ANDROID.keystore***)  
   Click the “OK” button  
   
3. Click Pipeline -> New Pipeline



1. Select Azure Repos Git



1. Select a repository from the list  
   
2. Select Android  
   
3. Now, write this code in a yml file,

Note:

1. Here we are triggering APK for two build variants **weUat** and **weProd,** so we have to mention ***assembleweuat*** *&* ***assembleweprod,*** respectively.
2. **epic-development** is the branch name

*trigger:*

*- epic-development*

*pool:*

*vmImage: 'macos-latest'*

*stages:*

*- stage: 'Build\_APK\_Artifacts'*

*jobs:*

*- job: 'Build\_UAT\_Process'*

*steps:*

*- task: Gradle@3*

*name: 'gradlewAssembleUat'*

*inputs:*

*workingDirectory: ''*

*gradleWrapperFile: 'gradlew'*

*gradleOptions: '-Xmx3072m'*

*javaHomeOption: 'JDKVersion'*

*jdkVersionOption: '1.17'*

*publishJUnitResults: false*

*testResultsFiles: '\*\*/TEST-\*.xml'*

*tasks: 'assembleweuat'*

*- task: AndroidSigning@3*

*inputs:*

*apkFiles: '\*\*/\*.apk'*

*apksignerKeystoreFile: 'BVN\_ANDROID.keystore'*

*apksignerKeystorePassword: '$(KeyStorePassword)'*

*apksignerKeystoreAlias: 'aliasName'*

*apksignerKeyPassword: '$(AliasPassword)'*

*- task: CopyFiles@2*

*inputs:*

*Contents: '\*\*/\*.apk'*

*TargetFolder: '$(build.artifactStagingDirectory)'*

*- task: PublishBuildArtifacts@1*

*inputs:*

*artifactName: UAT*

*- job: 'Build\_Prod\_Process'*

*dependsOn: Build\_UAT\_Process*

*steps:*

*- task: Gradle@3*

*name: 'gradlewAssembleProd'*

*inputs:*

*workingDirectory: ''*

*gradleWrapperFile: 'gradlew'*

*gradleOptions: '-Xmx3072m'*

*javaHomeOption: 'JDKVersion'*

*jdkVersionOption: '1.17'*

*publishJUnitResults: false*

*testResultsFiles: '\*\*/TEST-\*.xml'*

*tasks: 'assembleweprod'*

*- task: AndroidSigning@3*

*inputs:*

*apkFiles: '\*\*/\*.apk'*

*apksignerKeystoreFile: 'BVN\_ANDROID.keystore'*

*apksignerKeystorePassword: '$(KeyStorePassword)'*

*apksignerKeystoreAlias: 'aliasName'*

*apksignerKeyPassword: '$(AliasPassword)'*

*- task: CopyFiles@2*

*inputs:*

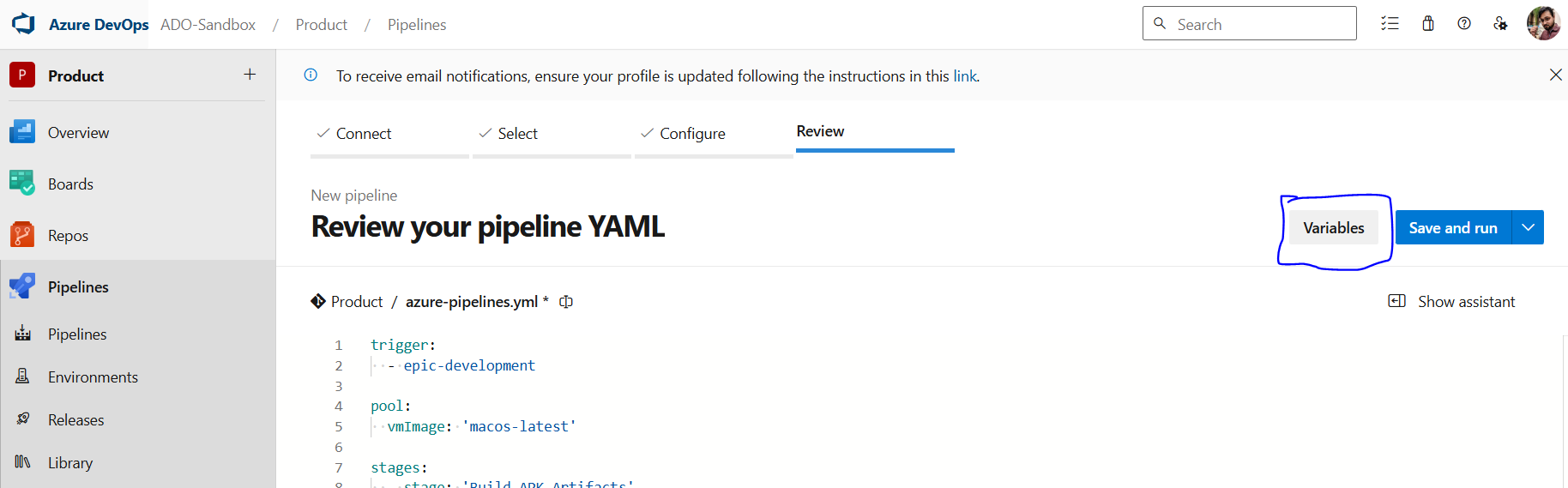
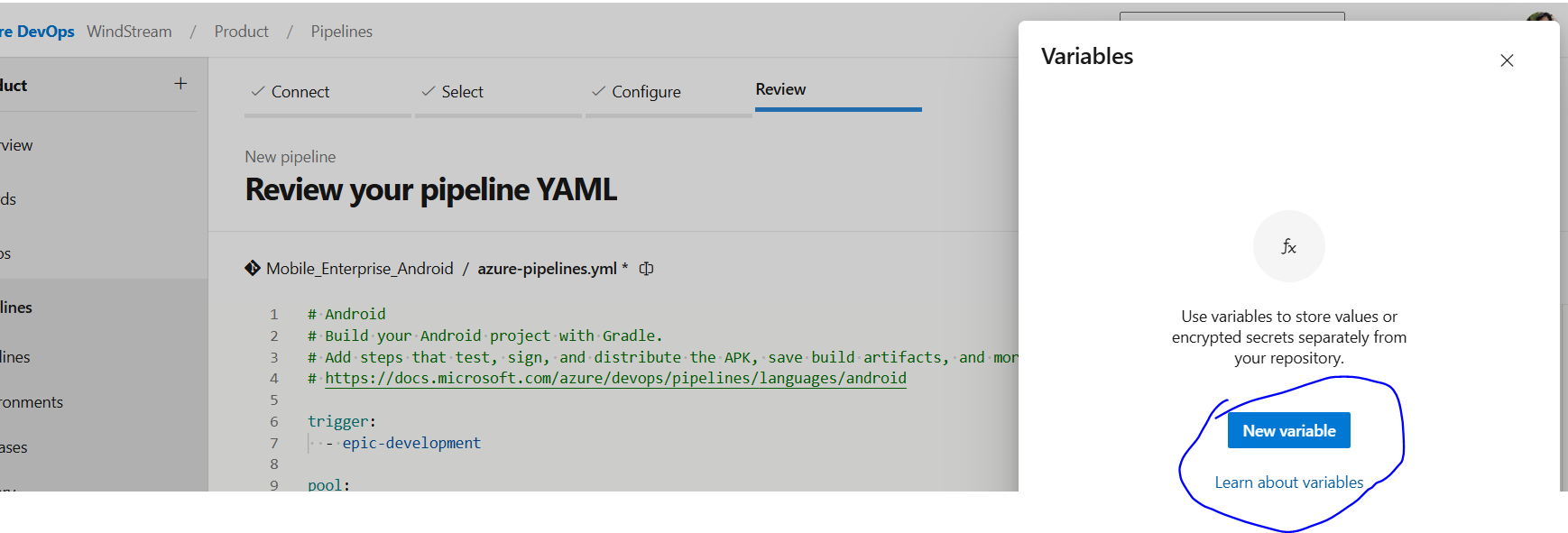
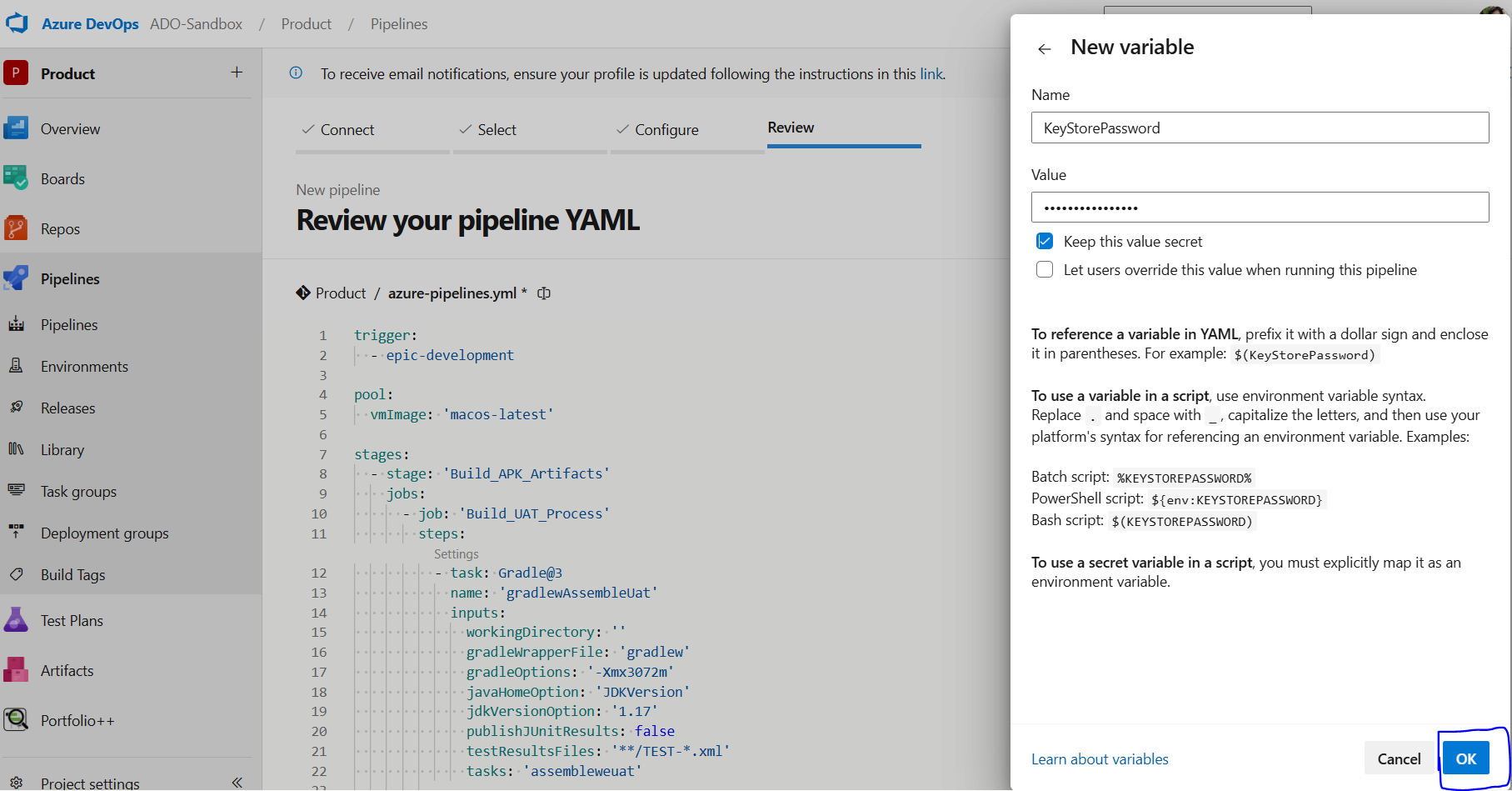
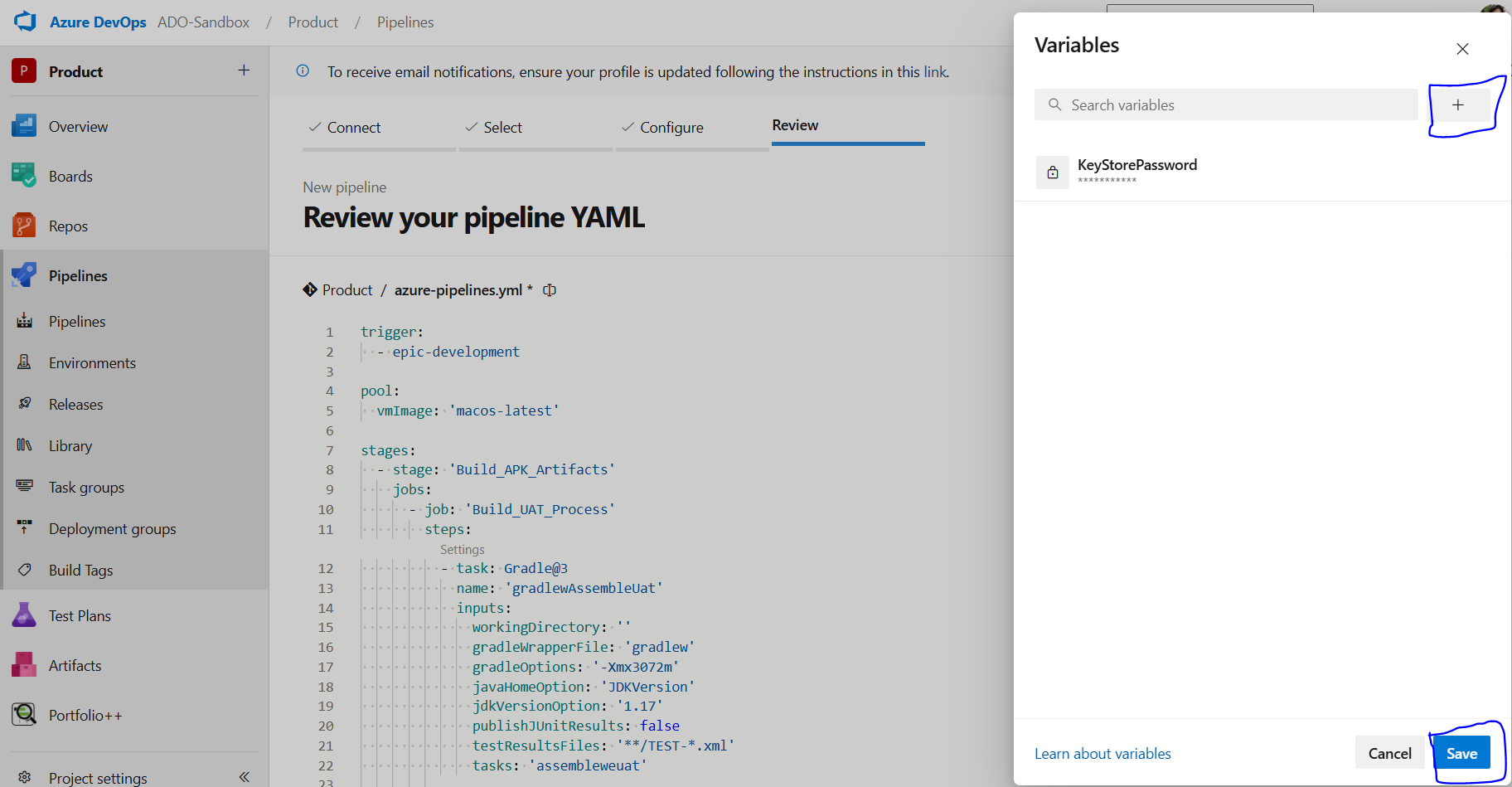
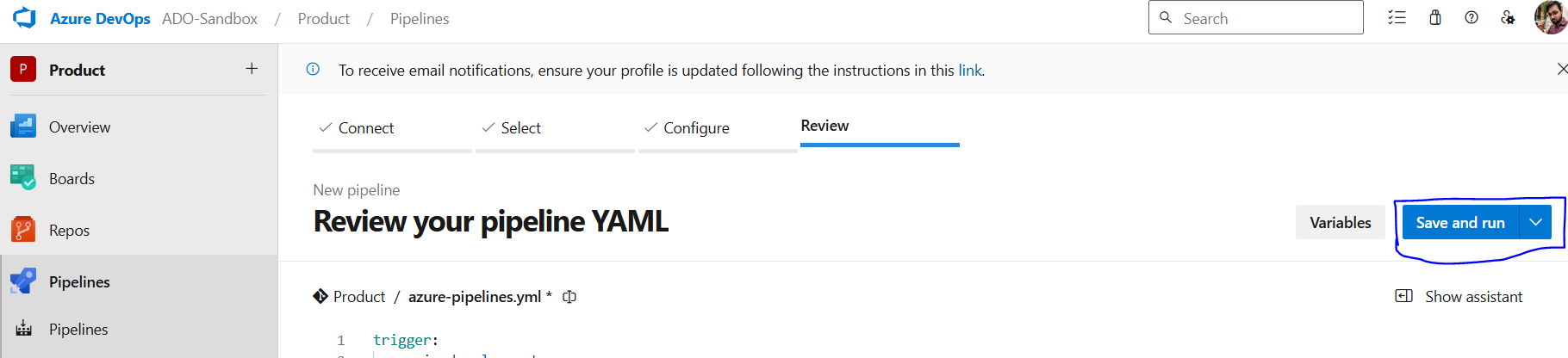
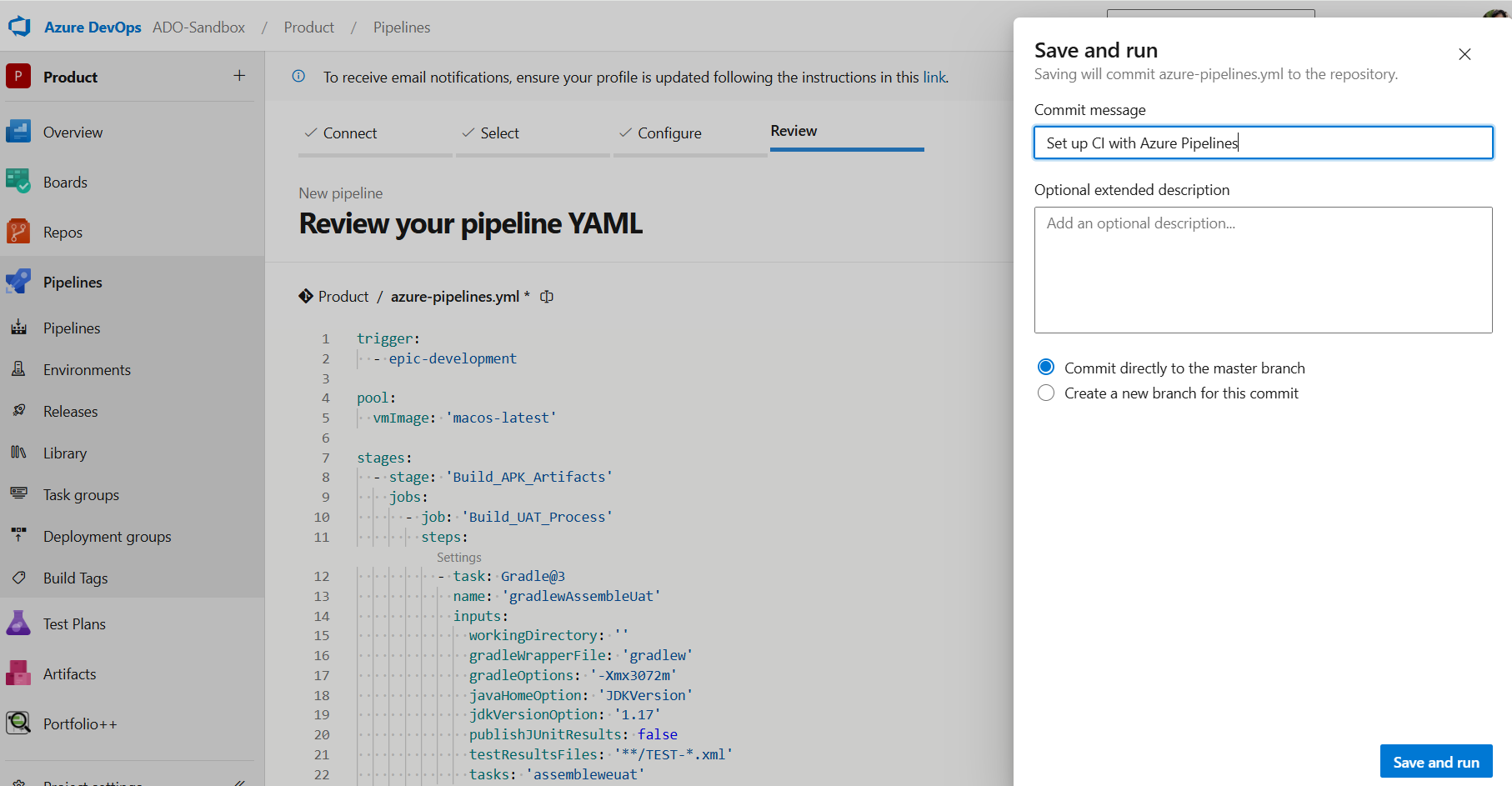
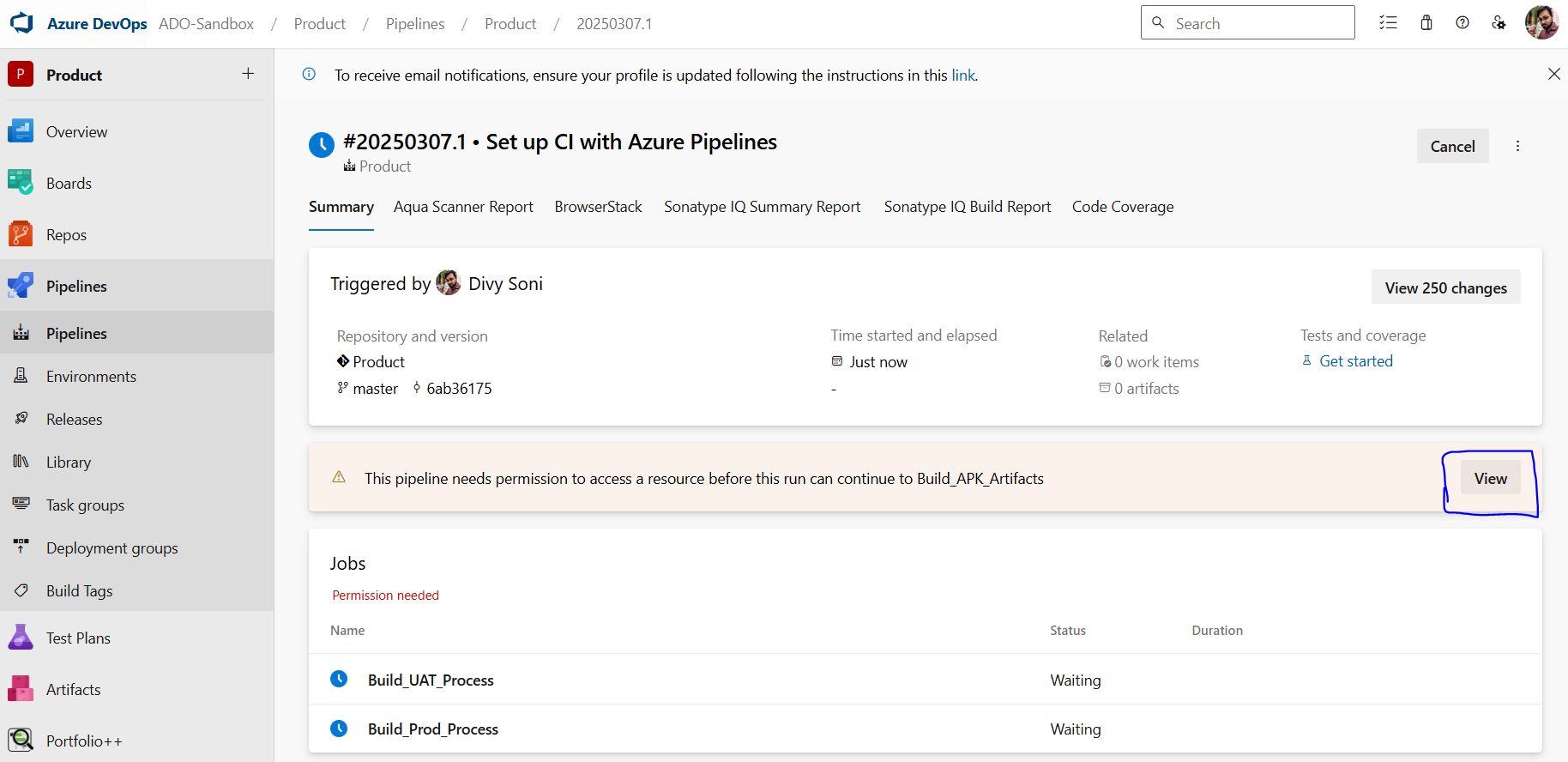
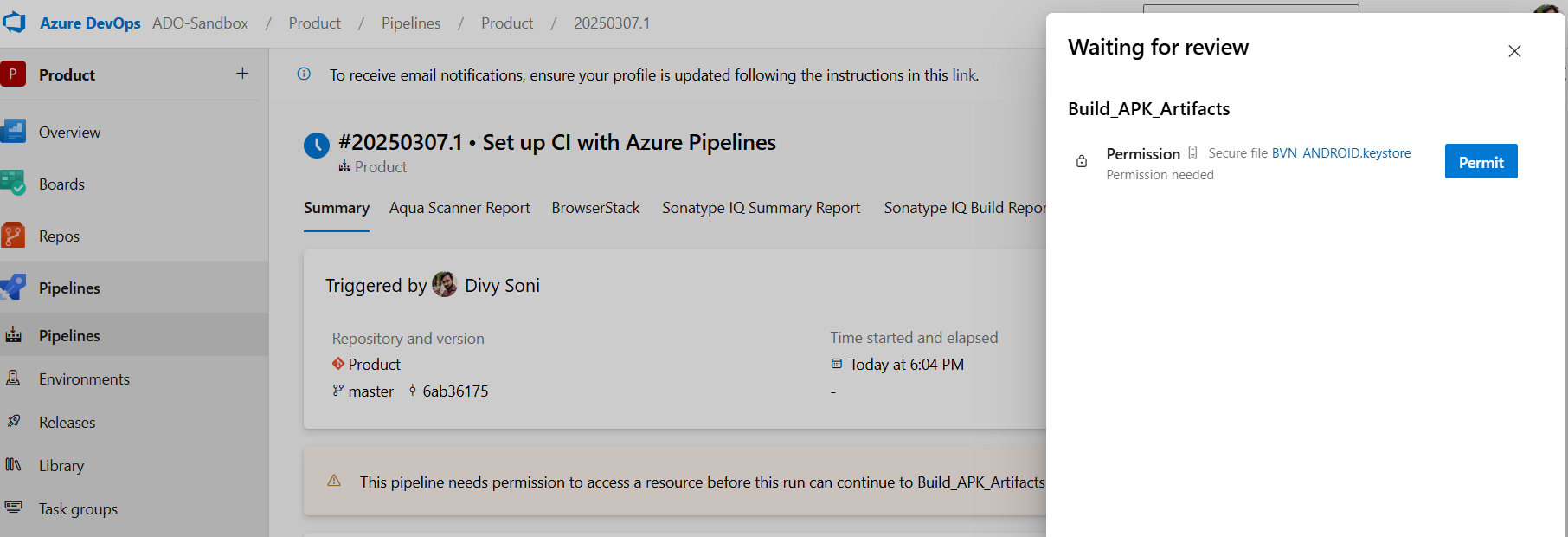
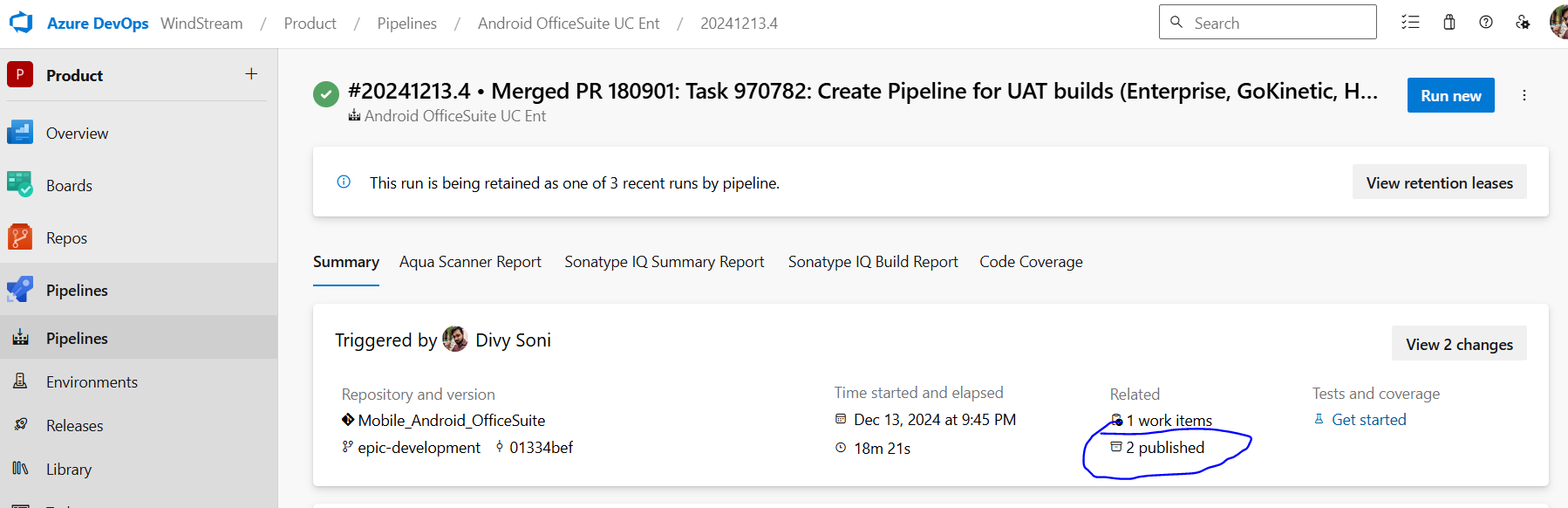
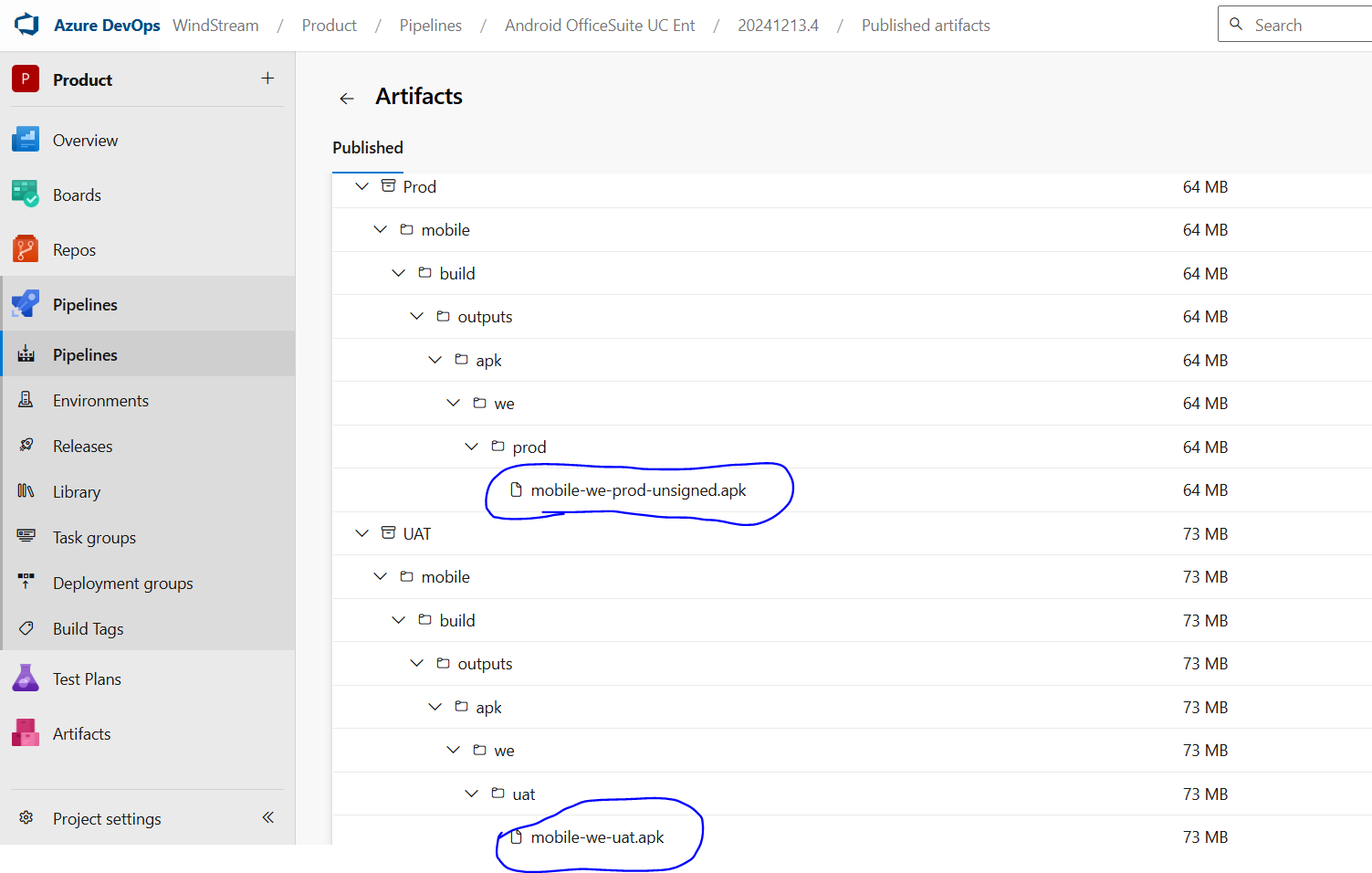
*Contents: '\*\*/\*.apk'*

*TargetFolder: '$(build.artifactStagingDirectory)'*

*- task: PublishBuildArtifacts@1*

*inputs:*

*artifactName: Prod*

1. Click on Variable  
   
2. Click on New Variable  
   
3. Name: *KeyStorePassword*  
   Value: <YourKeyStorePassword>  
   Select checkbox “Keep this value secret” to hide value  
   Click “OK” button  
   
4. Click “+” button  
   
5. Name: *AliasPassword*  
   Value: <YourAliasPassword>  
   Select checkbox “Keep this value secret” to hide value
6. Click “Save”. It will save Variables.
7. Now click “Save and run”   
   
8. Click “Save and run”  
   
9. Now, the build will trigger
10. After some time, this pop-up will appear. Then click on “View”.  
    
11. Now permit to access the keystore file by clicking on “Permit”.  
    
12. After the successful completion of the build, click on 2 published  
    
13. Now, UAT and Prod APKs are ready  
    

Creating AAB

To Create AAB, we need to write the below code in the yml file:

Note: Here we are triggering AAB for build variant **weProd,** so we have mentioned ***bundleweprod***.

trigger:

- epic-development

pool:

vmImage: 'macos-latest'

stages:

- stage: 'Build\_Release\_AAB\_Artifact'

jobs:

- job: 'Build\_Release\_AAB\_Process'

steps:

- task: DownloadSecureFile@1

name: KeyStoreFile

inputs:

secureFile: 'BVN\_ANDROID.keystore'

- powershell: 'echo $(KeyStoreFile.secureFilePath)'

displayName: 'PowerShell Script'

- task: Gradle@3

name: 'gradlewBuildAAB'

inputs:

workingDirectory: ''

gradleWrapperFile: 'gradlew'

gradleOptions: '-Xmx3072m'

javaHomeOption: 'JDKVersion'

jdkVersionOption: '1.17'

publishJUnitResults: false

testResultsFiles: '\*\*/TEST-\*.xml'

tasks: 'bundleweprod'

- task: CmdLine@2

displayName: 'Signing and aligning AAB file(s) mobile\build\outputs\bundle\weProd\mobile-we-prod.aab'

inputs:

script: 'jarsigner -verbose -sigalg SHA256withRSA -digestalg SHA-256 -keystore $(KeyStoreFile.secureFilePath) -storepass $(KeyStorePassword) -keypass $(AliasPassword) $(system.defaultworkingdirectory)/mobile/build/outputs/bundle/weProd/mobile-we-prod.aab aliasName'

- task: CopyFiles@2

inputs:

SourceFolder: '$(system.defaultworkingdirectory)/mobile/build/outputs/bundle/'

Contents: '\*\*/\*.aab'

TargetFolder: '$(build.artifactstagingdirectory)'

- task: PublishBuildArtifacts@1

inputs:

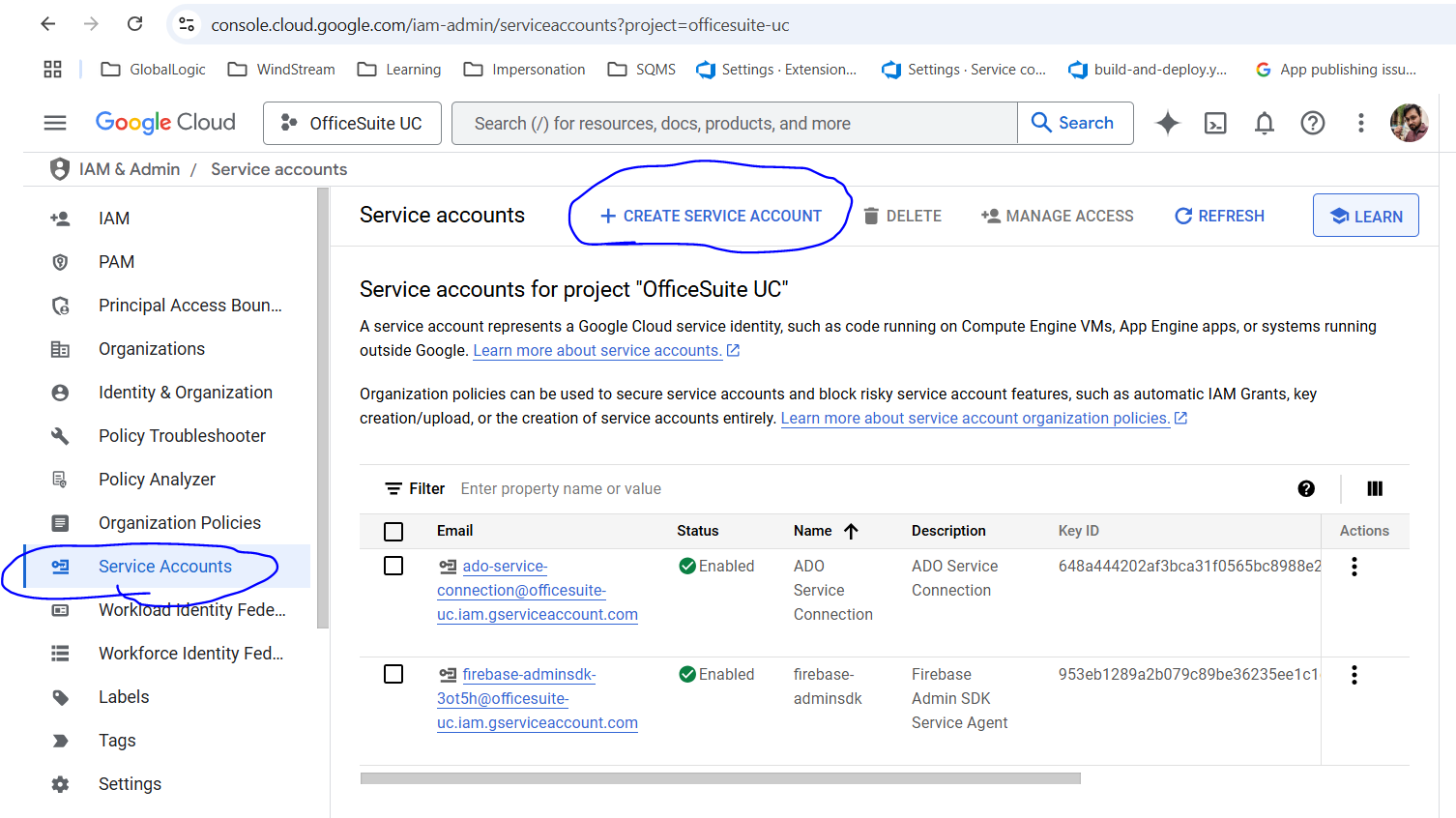
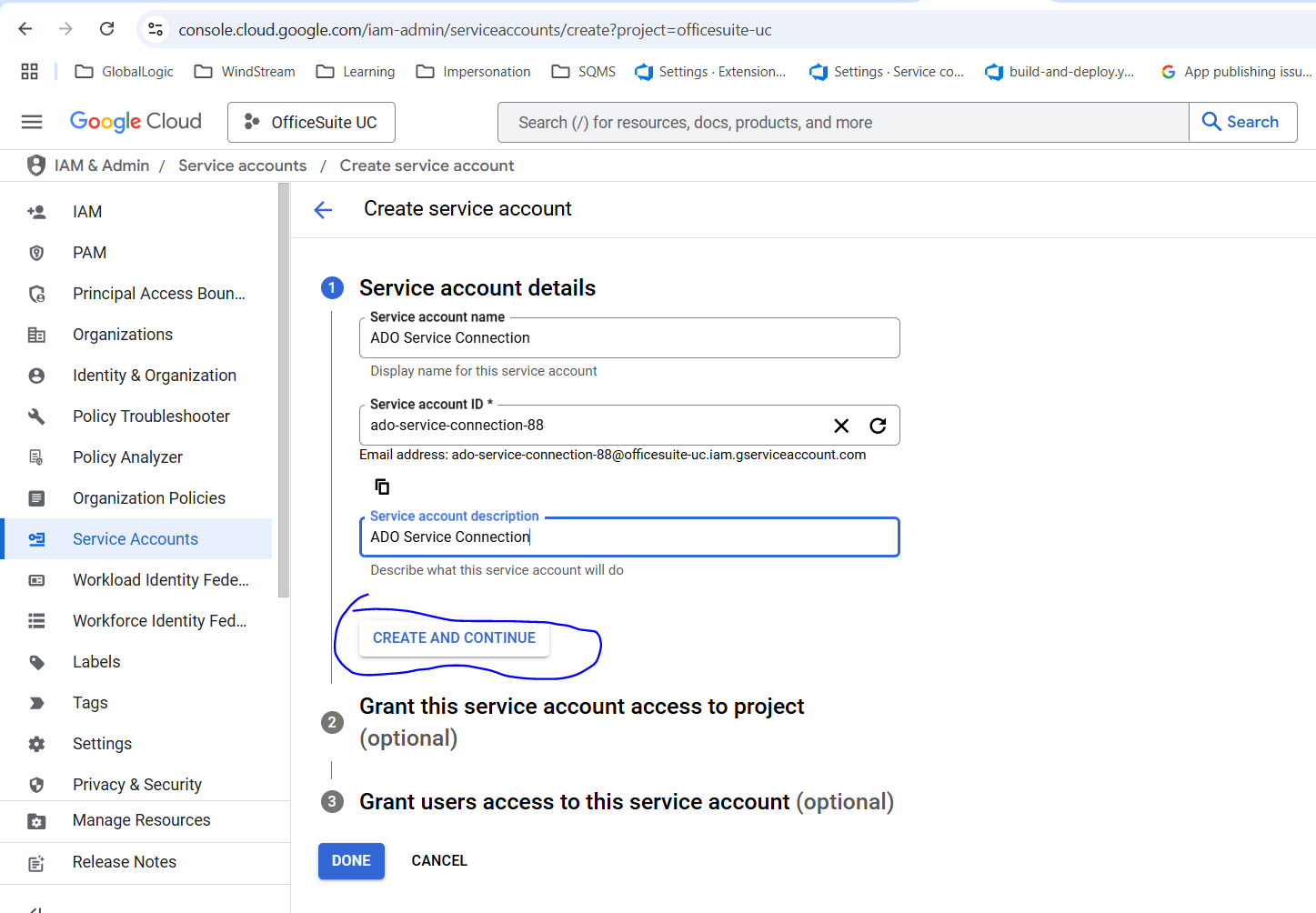
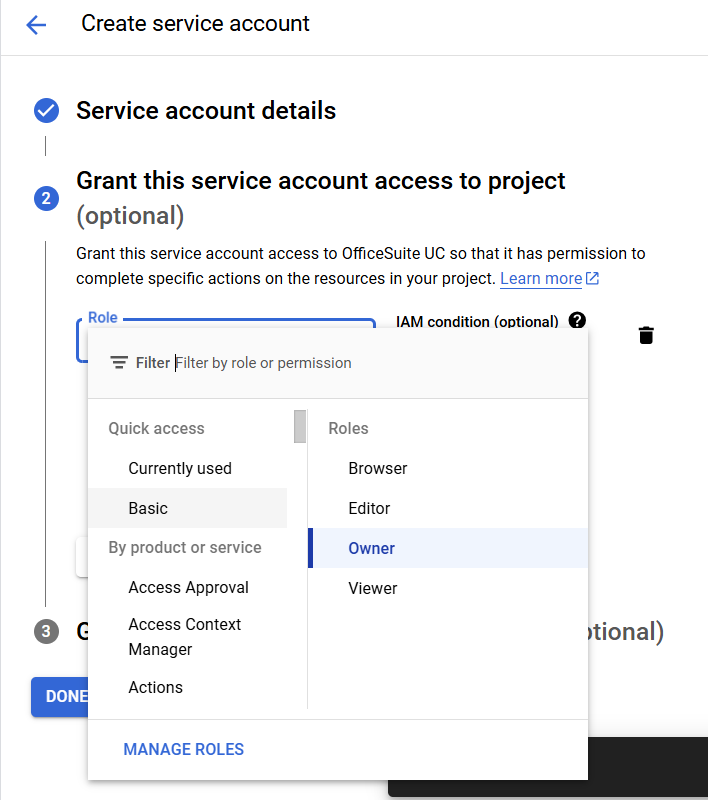
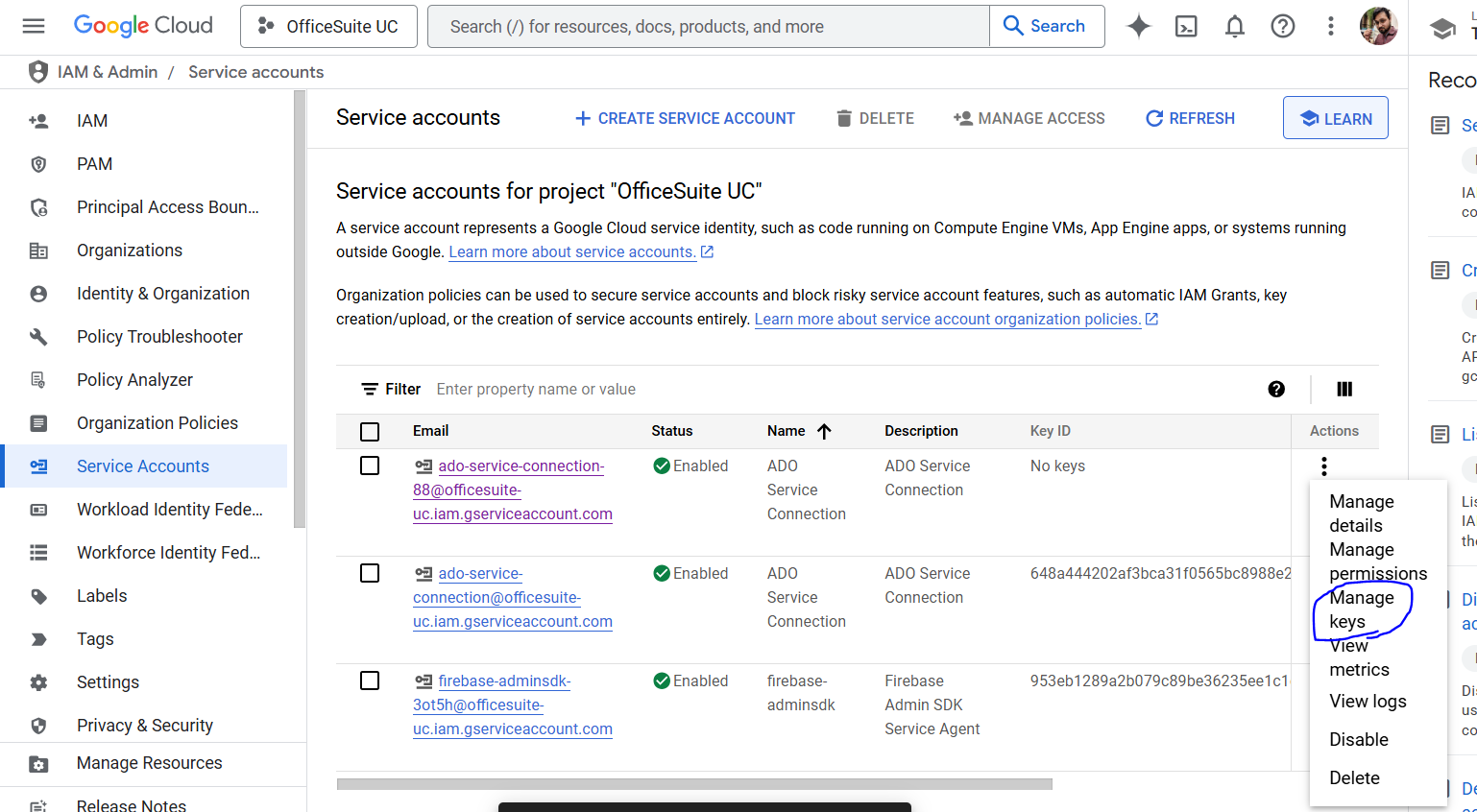
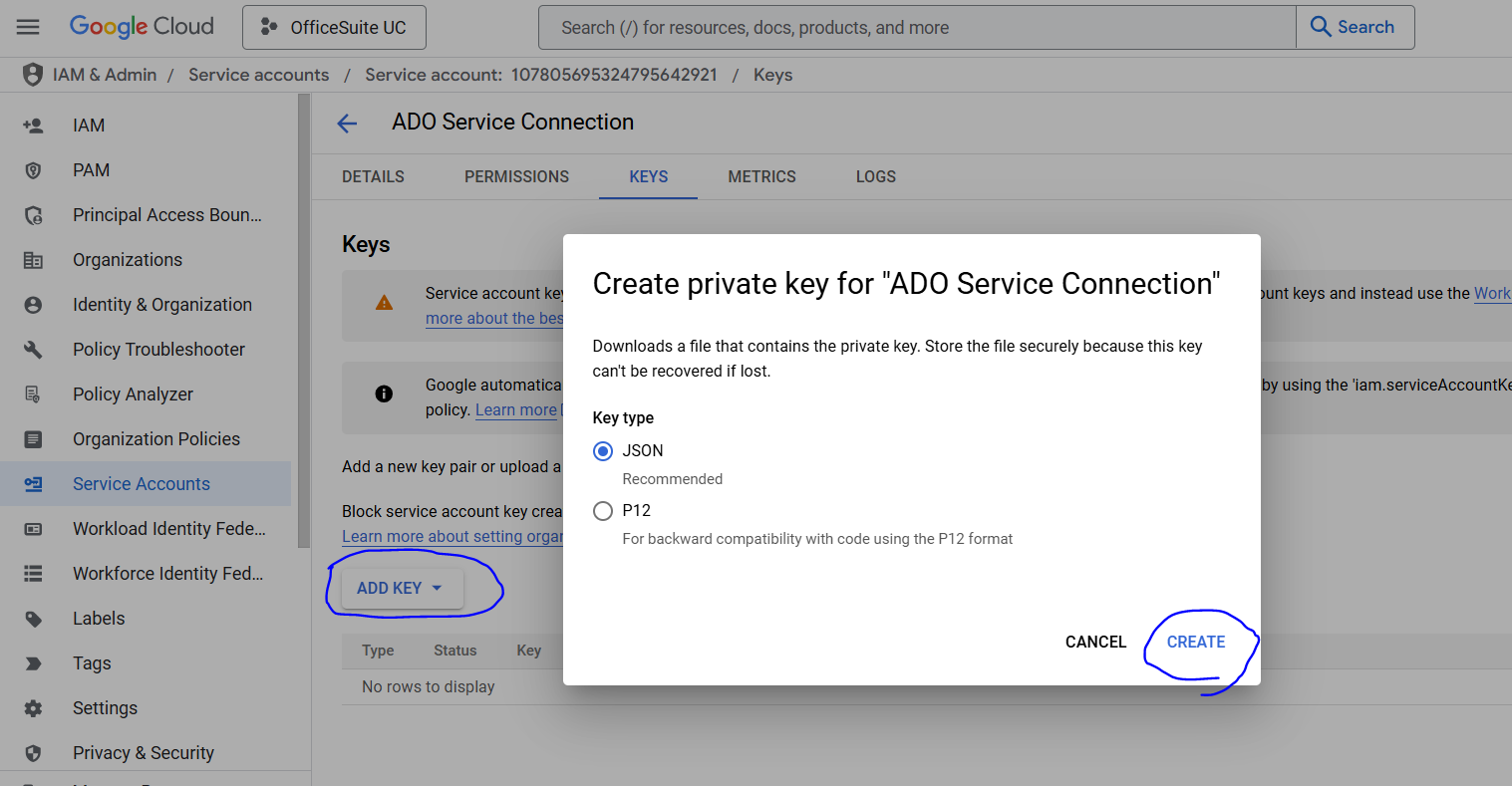
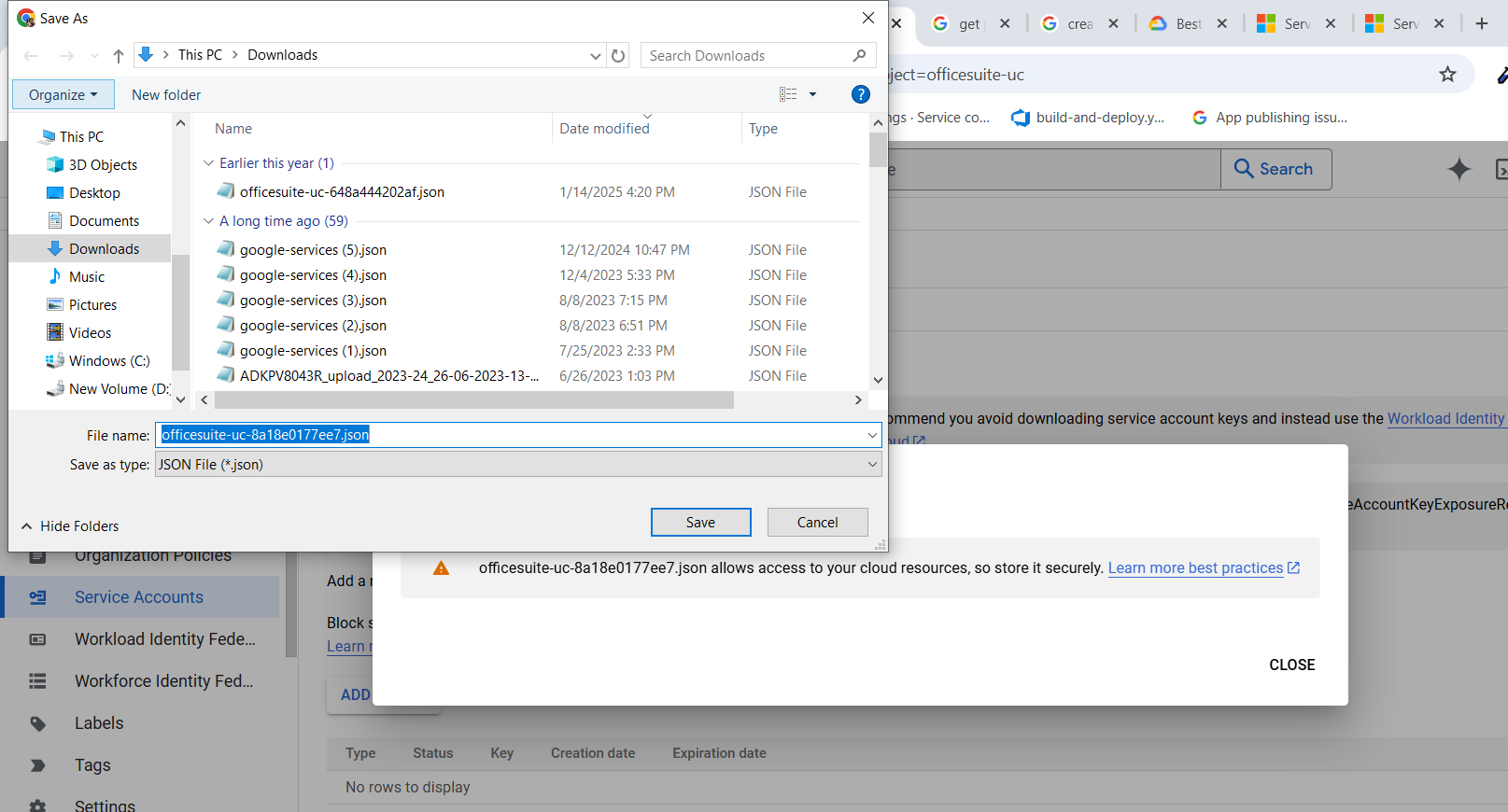
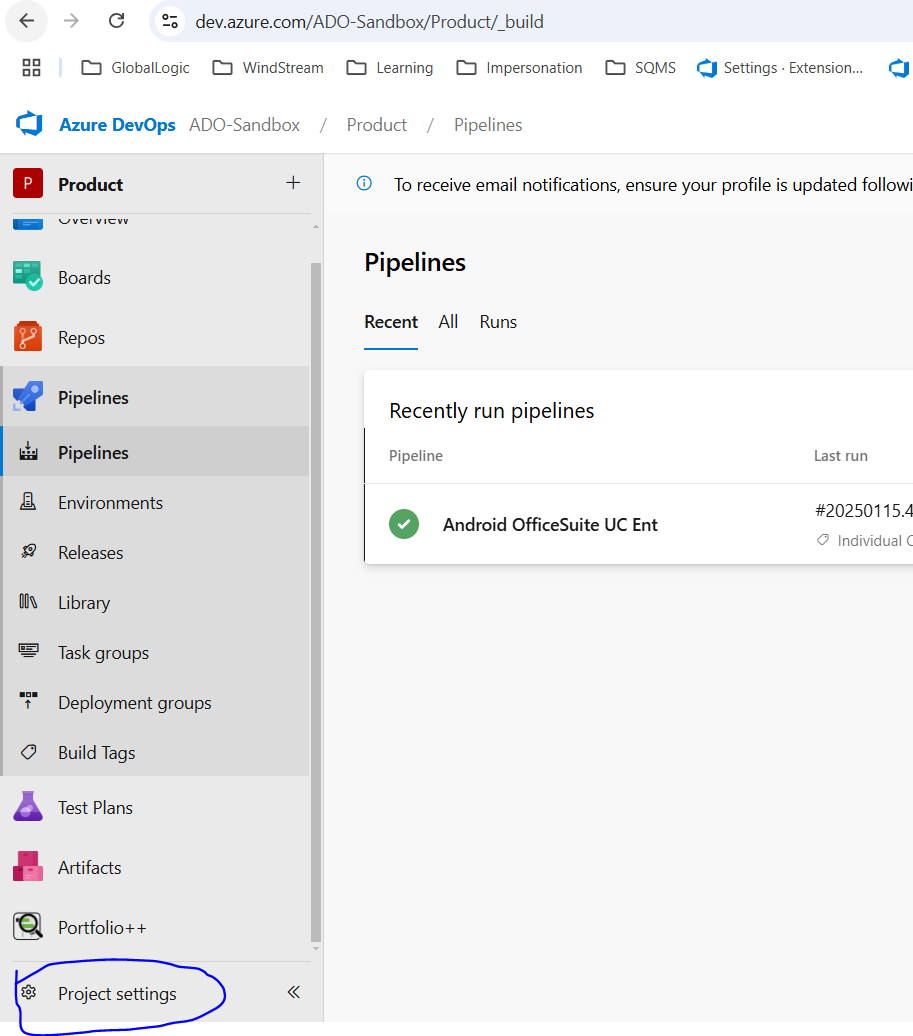
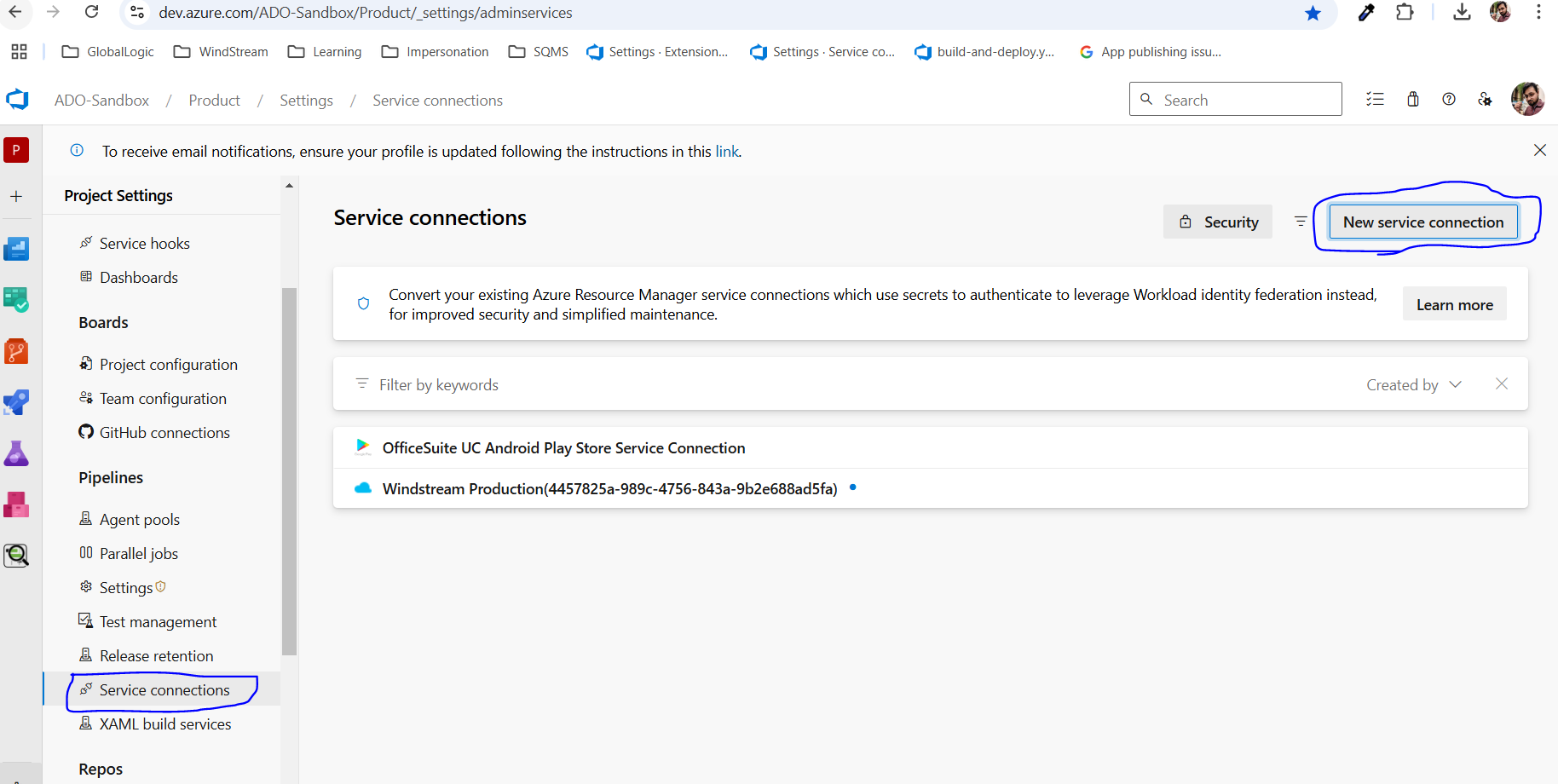
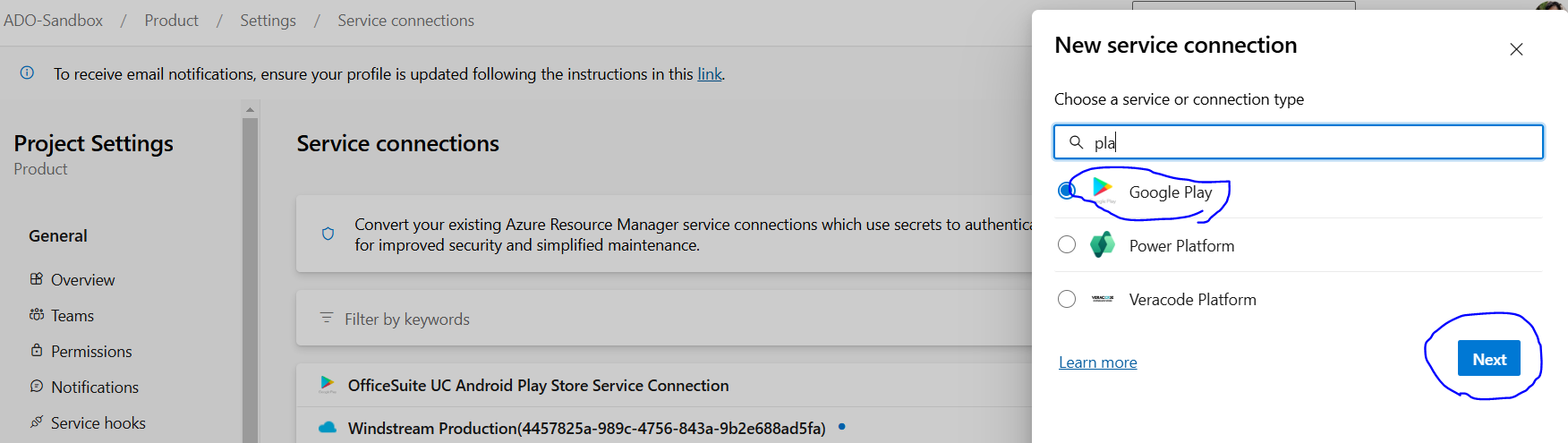
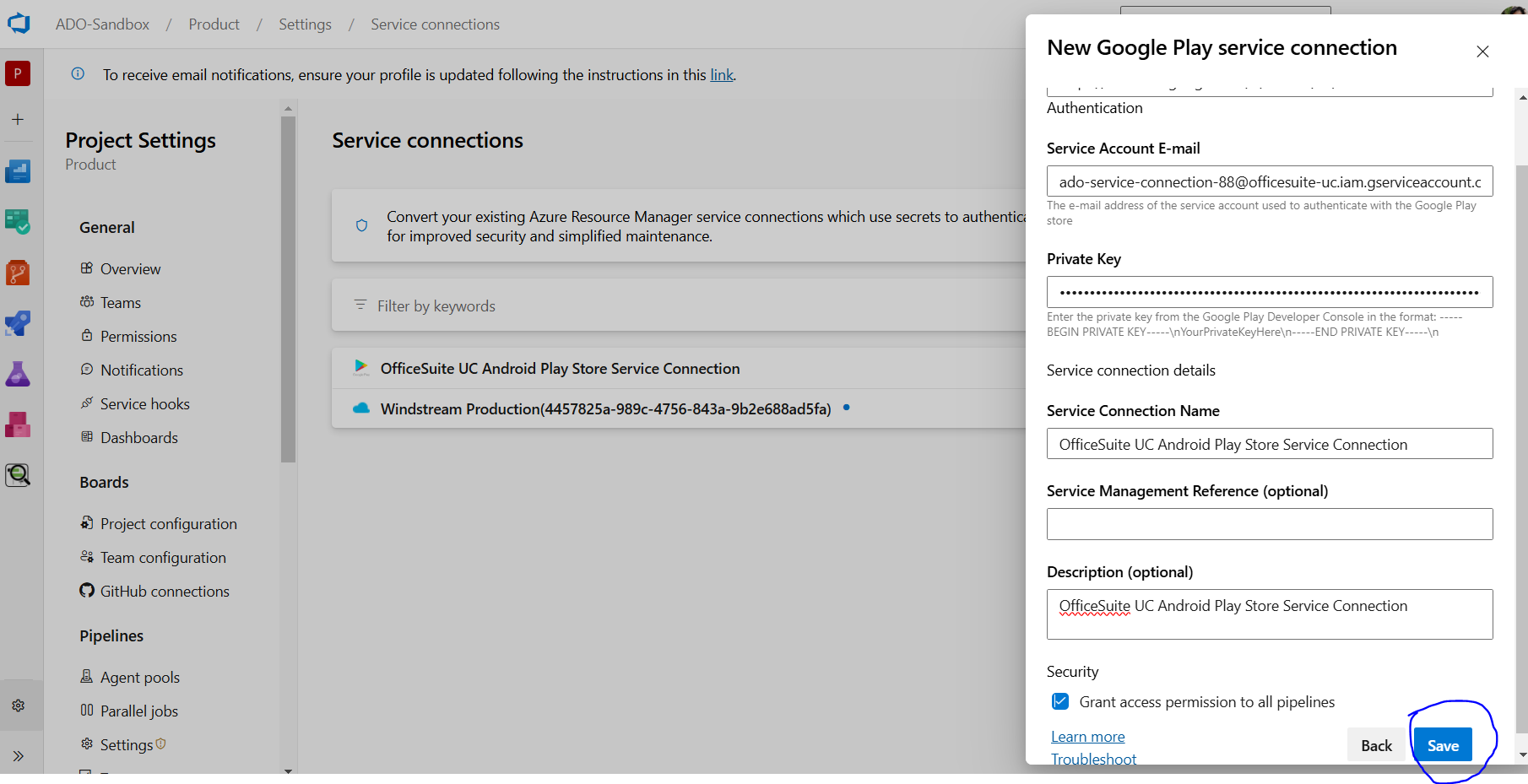
PathtoPublish: '$(Build.ArtifactStagingDirectory)'

artifactName: Prod

publishLocation: 'Container'

Publish to Play Store

To publish an app on Play Store:

1. Go to <https://console.cloud.google.com/iam-admin/iam>
2. Select Project
3. Click on “Service accounts” -> “Create Service Account”  
   
4. Fill in the details and click Create And Continue  
   
5. Select Role as “Owner”  
   
6. Click “Continue” -> “Done”
7. Click on Manage Keys  
   
8. Click on “Add Key” -> “Create”  
   
9. Save the JSON file in the computer  
   
10. Now go to Azure Devops and Click on Project Settings  
    
11. Go To Service Connection -> New Service Connection  
    
12. Search and select “Google Play” -> Click “Next”  
    
13. Enter the below details  
    Service Account E-mail: client\_email (from the JSON file we saved in Step 9)  
    Private Key: private\_key (from the JSON file we saved in Step 9)  
    Service Connection Name: Any name to identify the Service Connection  
    Click “Save”  
    Now, Service Connection has been saved.  
    
14. Add the below code in the yml file

*- task: GooglePlayRelease@4*

*inputs:*

*serviceConnection: 'OfficeSuite UC Android Play Store Service Connection'*

*applicationId: 'com.windstream.officesuite'*

*action: 'SingleBundle'*

*bundleFile: '$(build.artifactstagingdirectory)/\*\*/\*.aab'*

*track: 'internal'*

Note:   
 serviceConnection: Your Service Connection Name  
 applicationId: Your Application ID (Package name)  
 bundleFile: Location of AAB file

OR

Go to the YML file in the pipeline and enter all the details and click “Add”; it will write all these lines automatically

