**Integration Github with Azure Pipelines**

**Overview**

**GitHub** hosts over 100 million repositories containing applications of all shapes and sizes. But GitHub is just a start—those applications still need to get built, released, and managed to reach their full potential.

[Azure Pipelines](https://azure.microsoft.com/services/devops/pipelines/) that enables you to continuously build, test, and deploy to any platform or cloud. It has cloud-hosted agents for Linux, macOS, and Windows; powerful workflows with native container support; and flexible deployments to Kubernetes, VMs, and serverless environments.

Azure Pipelines provides unlimited CI/CD minutes and 10 parallel jobs to every GitHub open source project for free. All open source projects run on the same infrastructure that our paying customers use. That means you’ll have the same fast performance and high quality of service. Many of the top open source projects are already using Azure Pipelines for CI/CD, such as Atom, CPython, Pipenv, Tox, Visual Studio Code, and TypeScript—and the list is growing every day.

In this lab, you’ll see how easy it is to set up **Azure Pipelines** with your **GitHub** projects and how you can achieve an end-to-end traceability from work items to code change, commit, to build and release.

**Setups**

**ARM Outputs extension**

<https://marketplace.visualstudio.com/items?itemName=keesschollaart.arm-outputs>

**Git installed from**

<https://git-scm.com/downloads>

**Visual Studio Code**

<https://code.visualstudio.com>

**Azure Pipelines extension for Visual Studio Code**

<https://marketplace.visualstudio.com/items?itemName=ms-azure-devops.azure-pipelines>

**GitHub Pull Requests extension for Visual Studio Code**

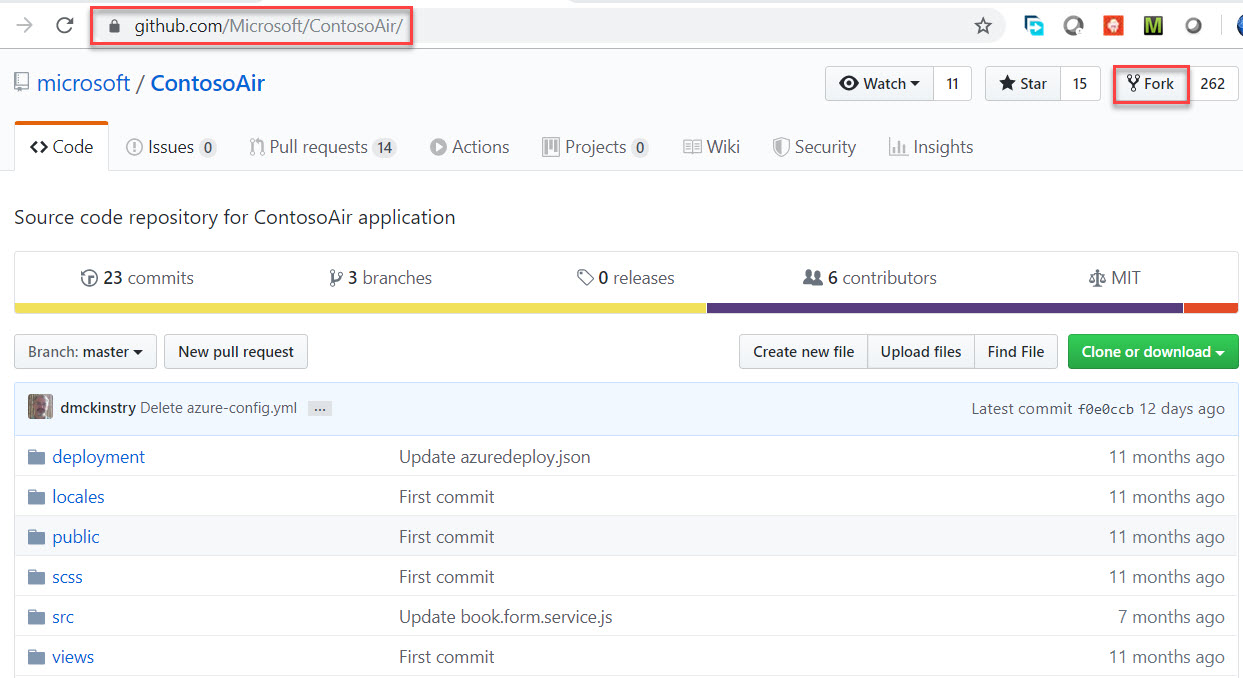
<https://marketplace.visualstudio.com/items?itemName=GitHub.vscode-pull-request-github>

**Load Sample Project**

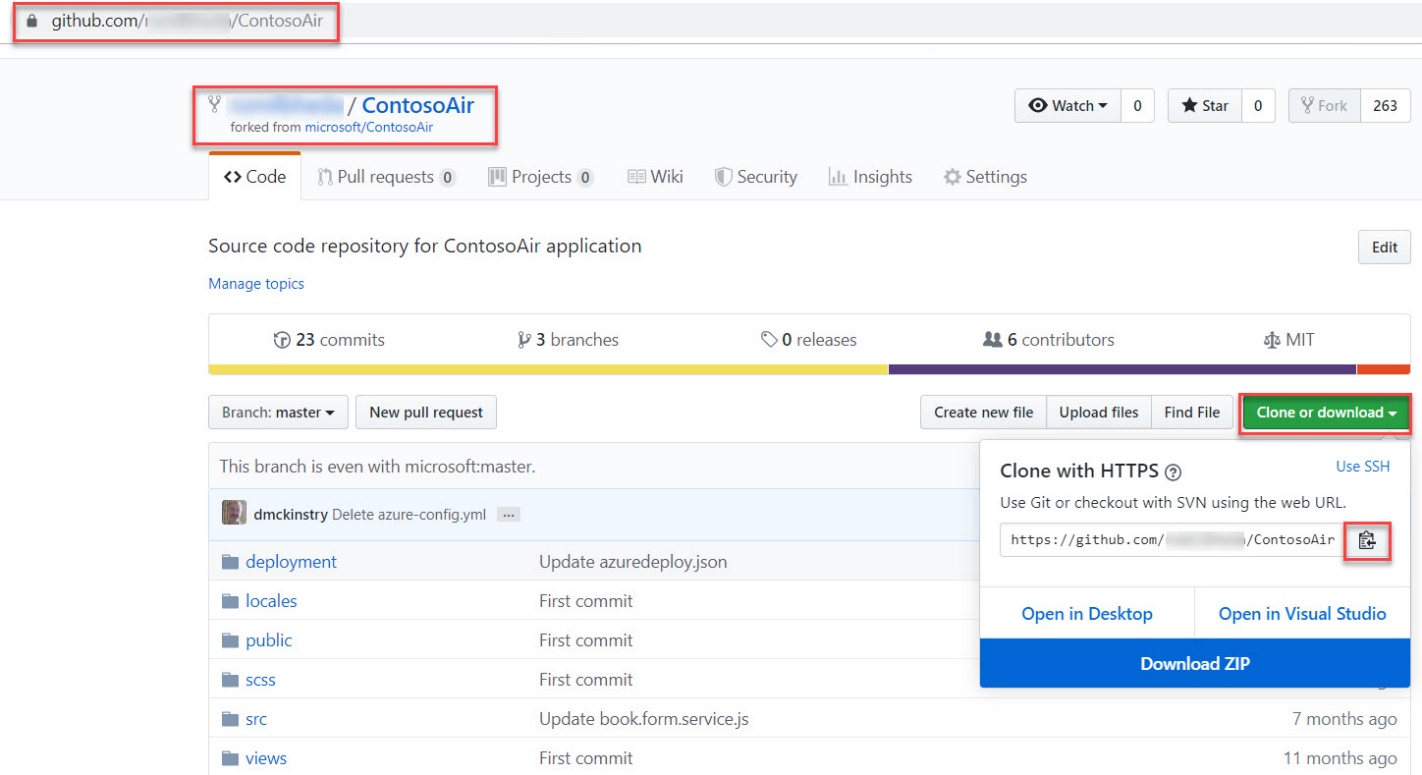
Open <https://github.com> and sign in with your credentials

Open <https://github.com/Microsoft/ContosoAir/>

Click Fork option

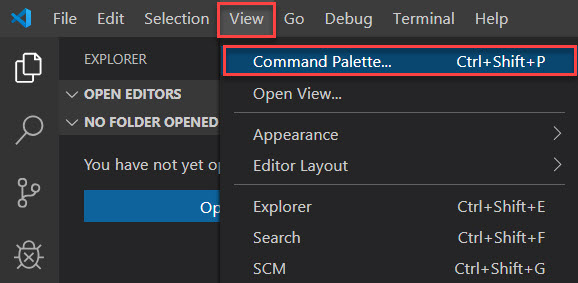


Click on Clone or download button -> Click on Copy

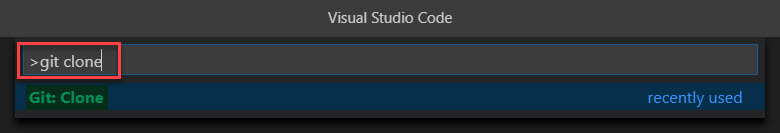


Start Visual Studio Code

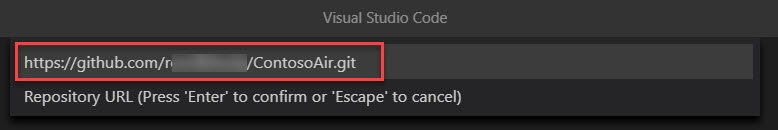
View -> Command Palette… option



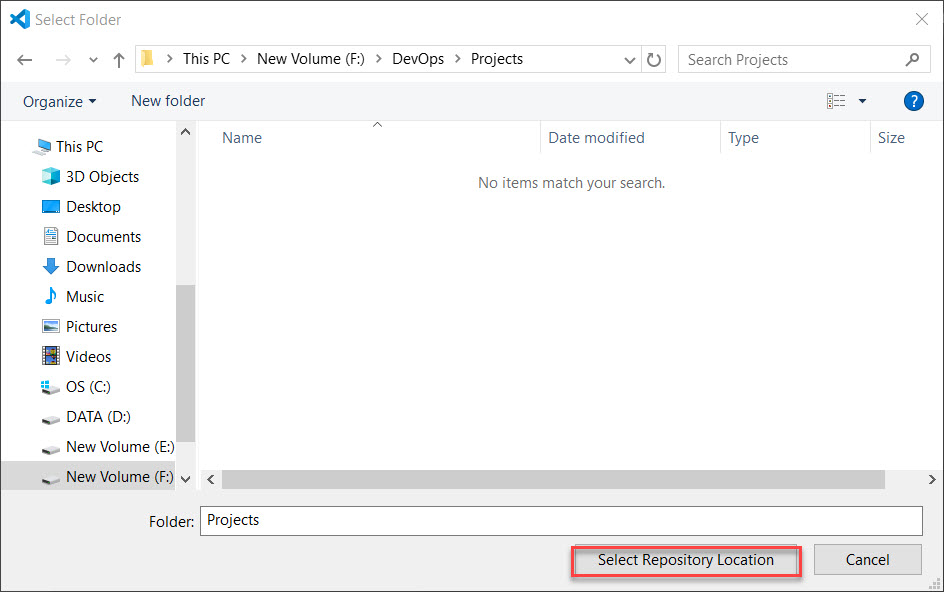
Type **git clone** and click on result



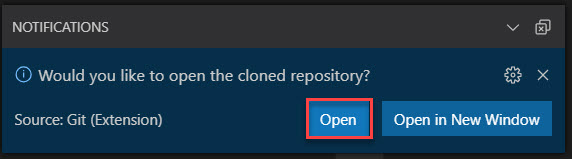
Paste that Github repo link and press enter



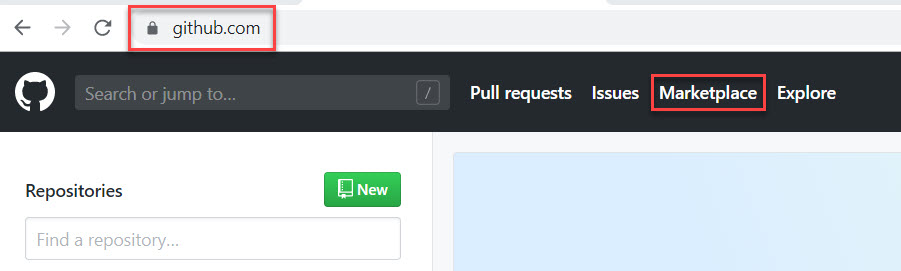
Select your repo location in local system.



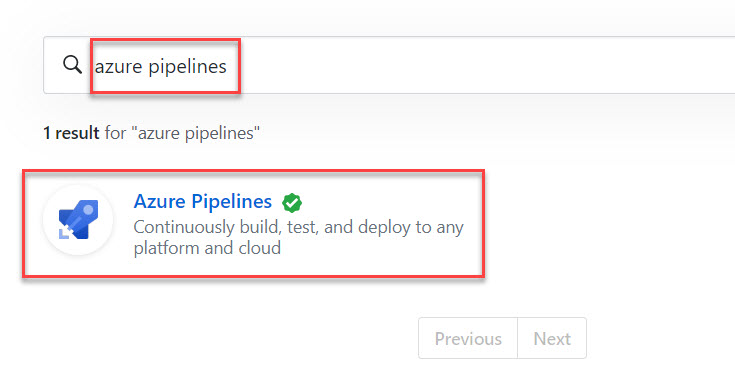
Click on Open button



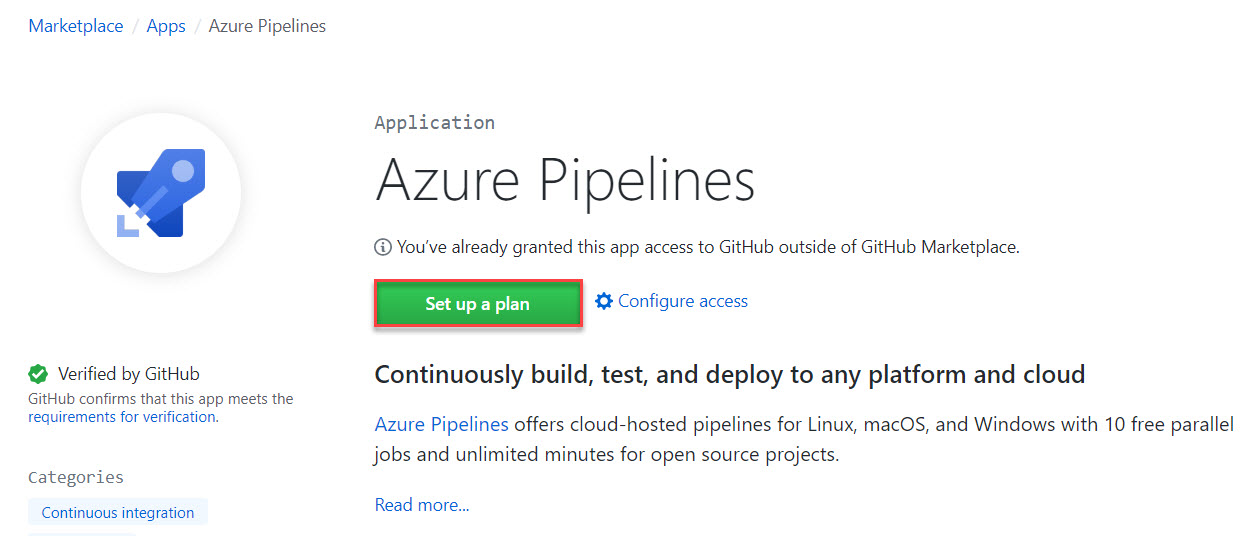
Navigate to github.com and select Marketplace option



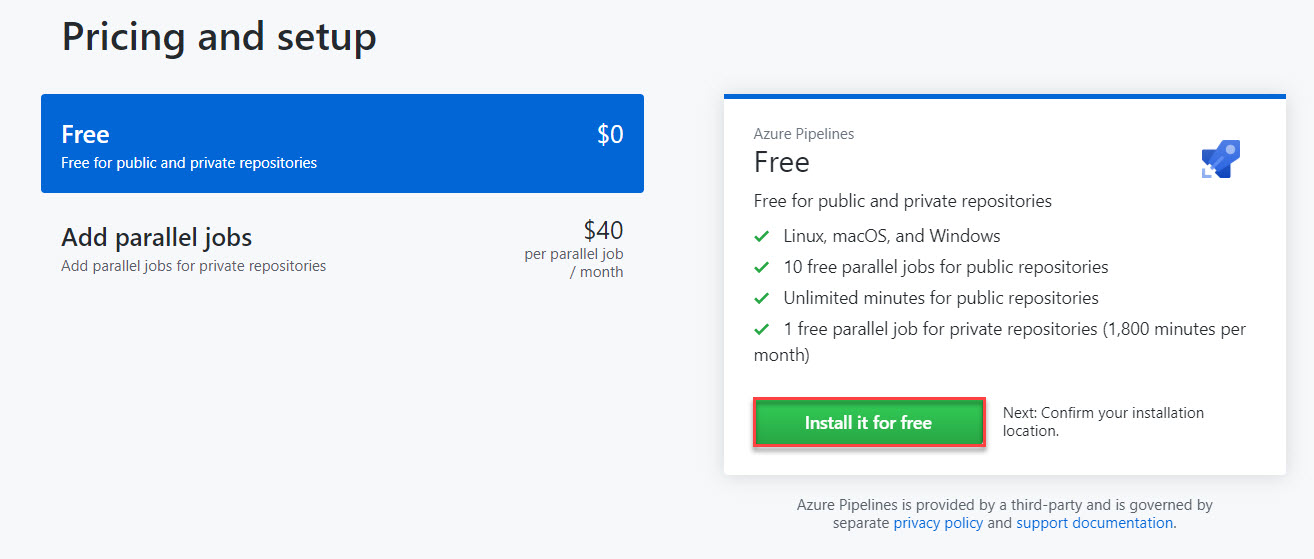
Search for “azure pipelines” and select Azure Pipelines



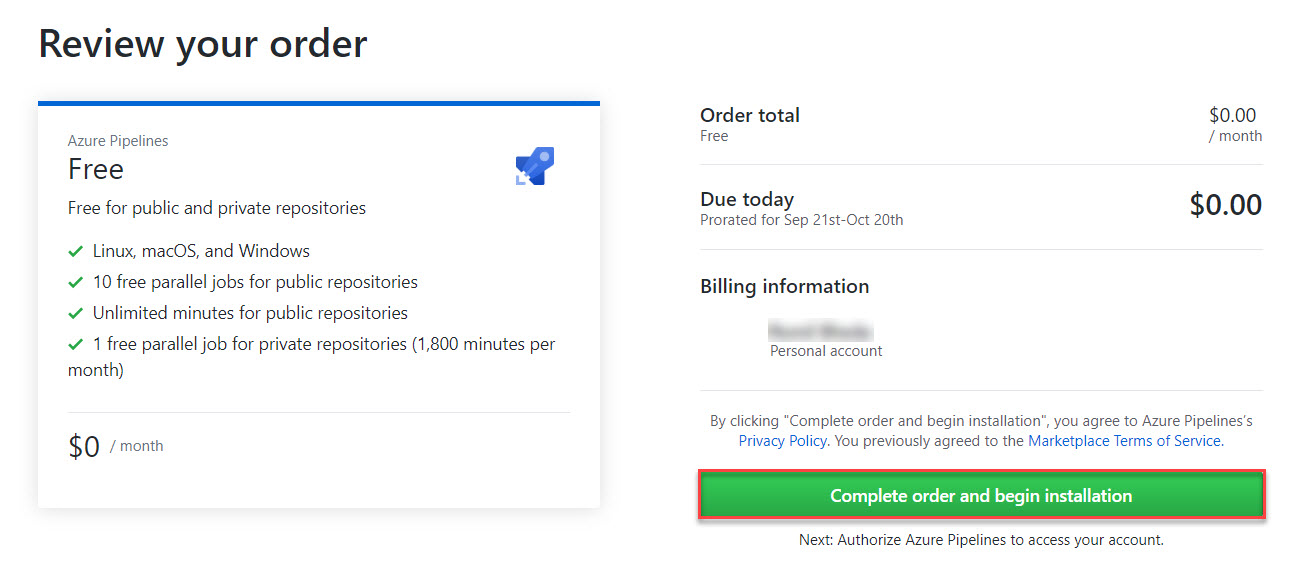
Click on **Set up a plan** button



Click on **Install it for free**

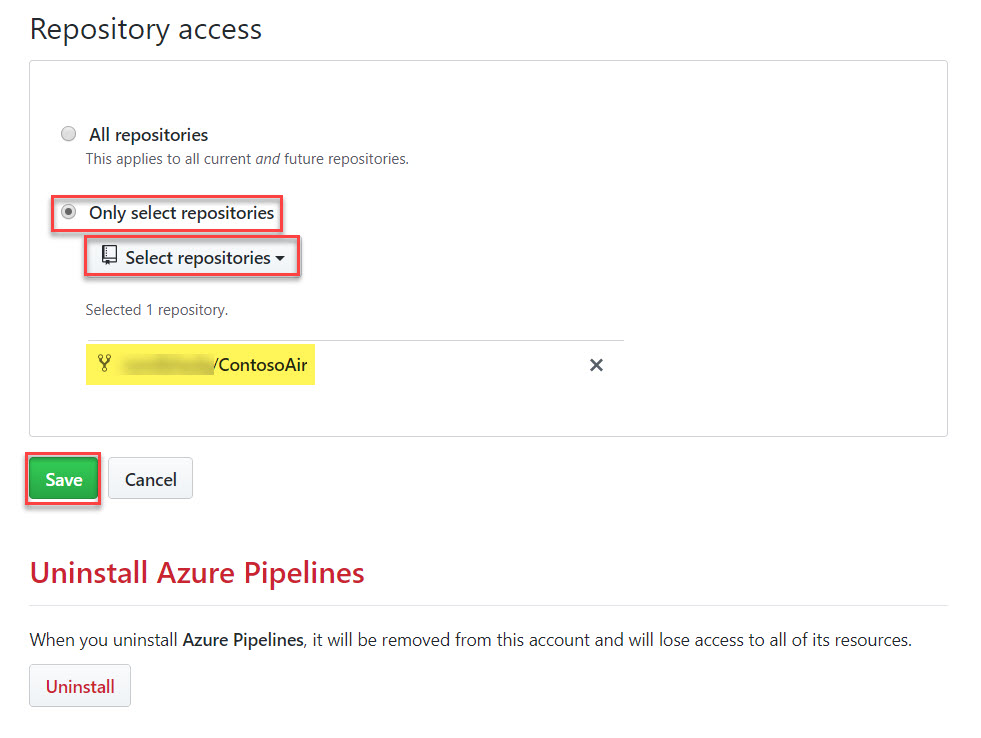


Click on **Complete order and begin installation** button

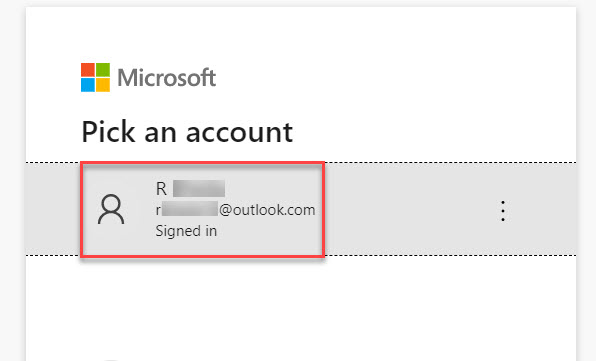


Select your repository Ex. **ContosoAir**

Click on **Save** button

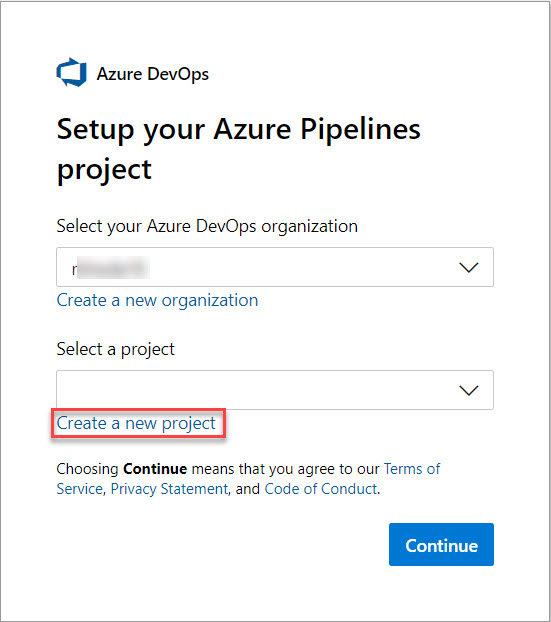


Enter Credentials



Select Organization from list

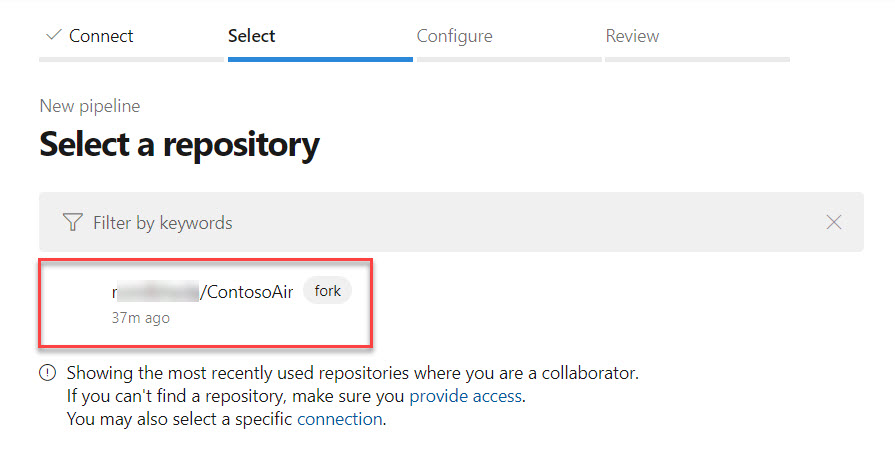
Click on **Create a new project** option



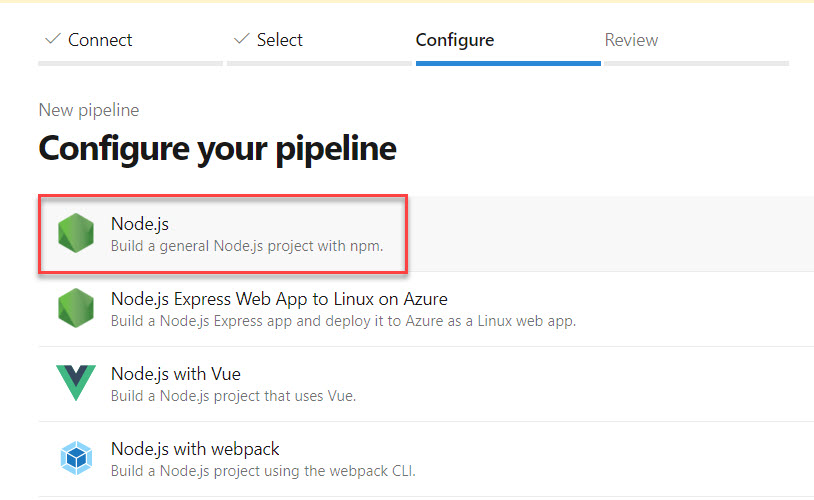
Enter Project Name: MyContosoAirProject



Select ContosoAir repo



Select Node.js



Update with below code:

pool:

vmImage: 'ubuntu-16.04'

trigger:

- master

steps:

- task: CopyFiles@2

displayName: 'Copy Files to: $(build.artifactstagingdirectory)/Templates'

inputs:

SourceFolder: deployment

Contents: '\*.json'

TargetFolder: '$(build.artifactstagingdirectory)/Templates'

- task: Npm@1

displayName: 'npm custom'

inputs:

command: custom

verbose: false

customCommand: 'install --production'

- task: ArchiveFiles@2

displayName: 'Archive $(Build.SourcesDirectory)'

inputs:

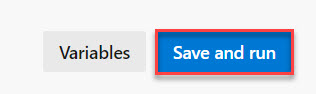
rootFolderOrFile: '$(Build.SourcesDirectory)'

includeRootFolder: false

- task: PublishBuildArtifacts@1

displayName: 'Publish Artifact: drop'

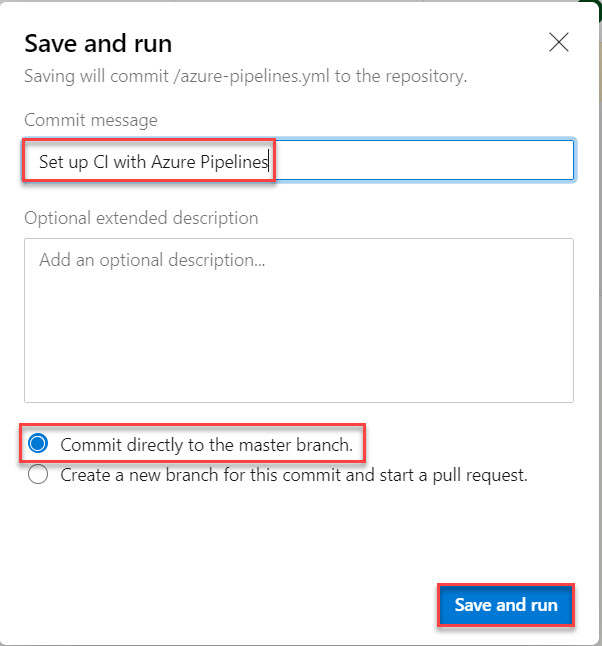
Click on Save and run button.



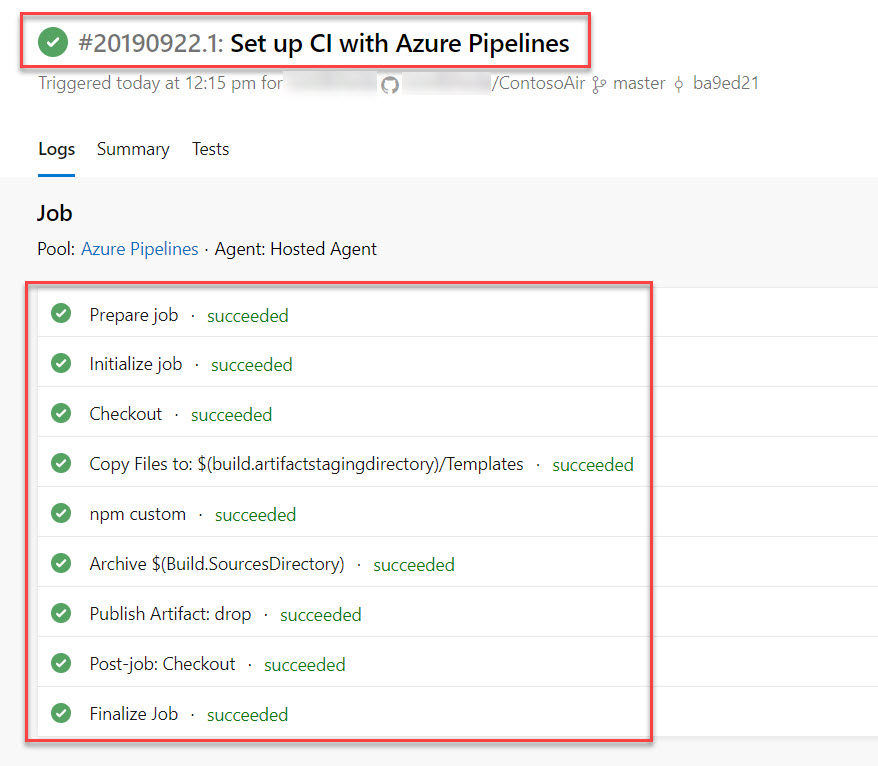
Commit message: Set up CI with Azure Pipelines

Select **Commit directly to the master branch.**

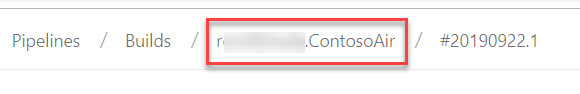
Click on **Save and Run** button



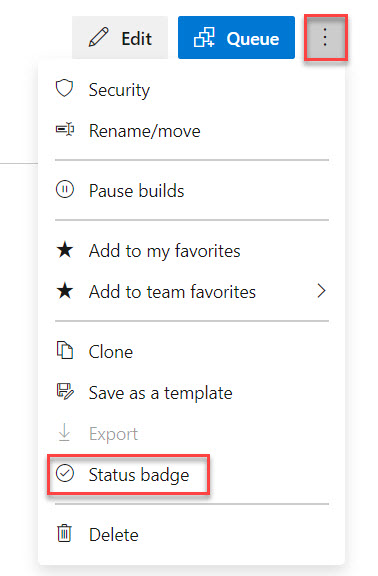
Wait for few minutes to build the project



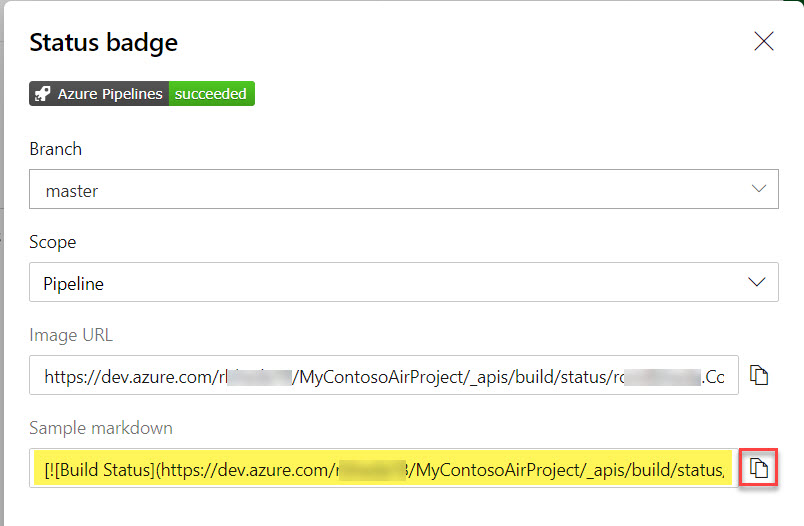
Click on Build Ex. abc.ContosoAir



Click on Vertical dots and Select **Status Badge** option



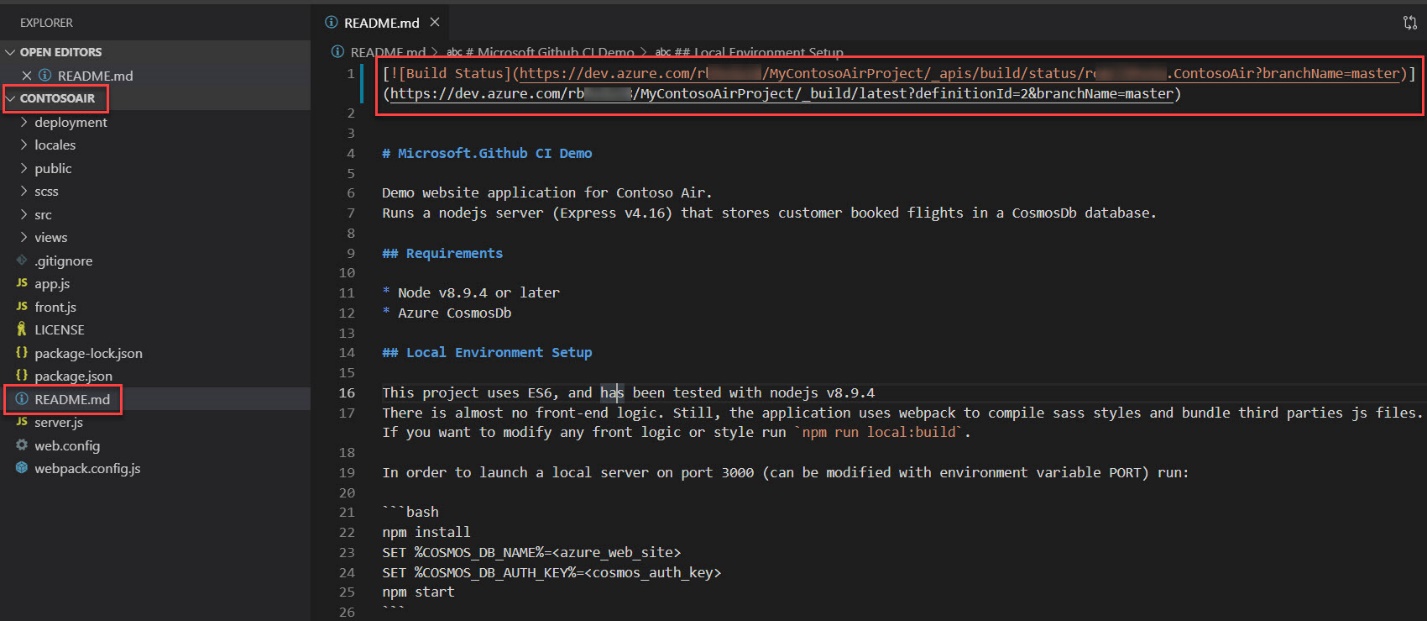
Check Sample markdown and copy



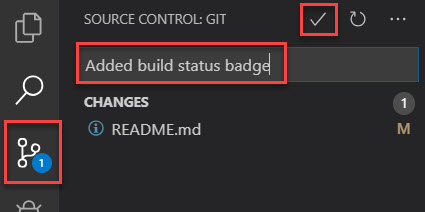
Navigate to Visual Studio Code

Expand ContosoAir Project and open README.md file

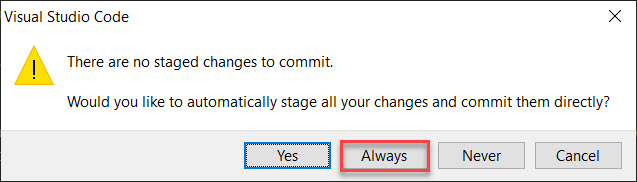
Paste badge code



Click on Source Control and write **Added build status badge**

Click on **Tick icon**

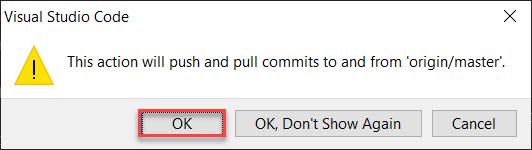
Click on **Always** button

\

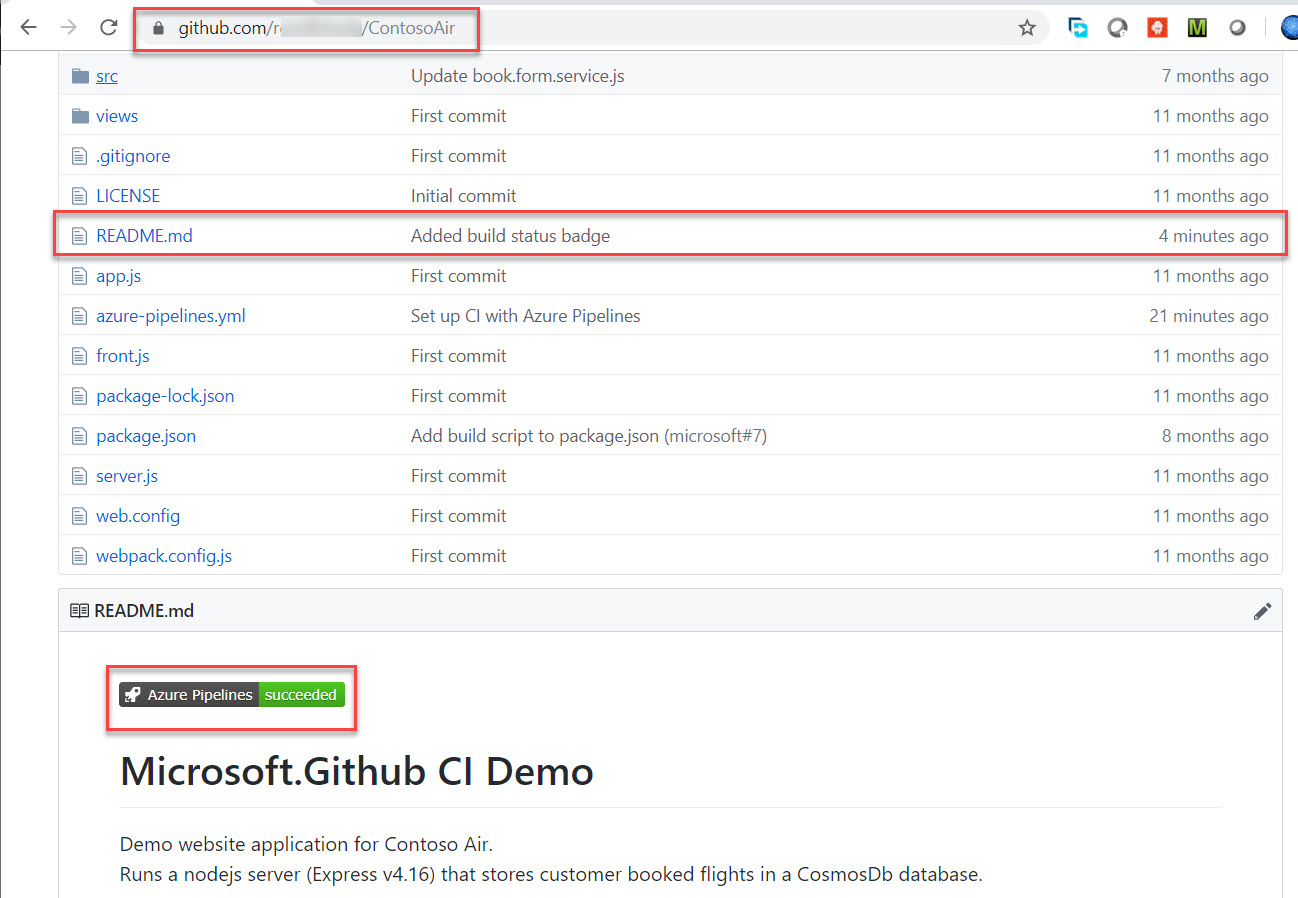
Check bottom-left side of Visual Studio code and click on Changes icons



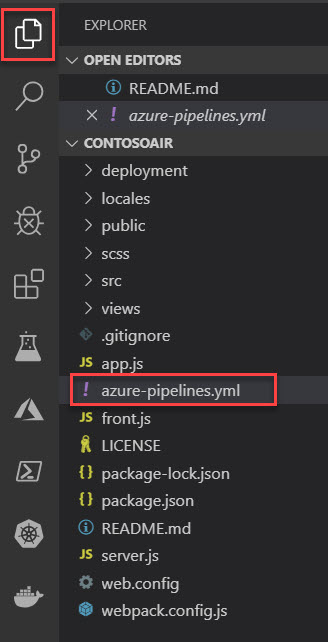
Click on Ok button to Push



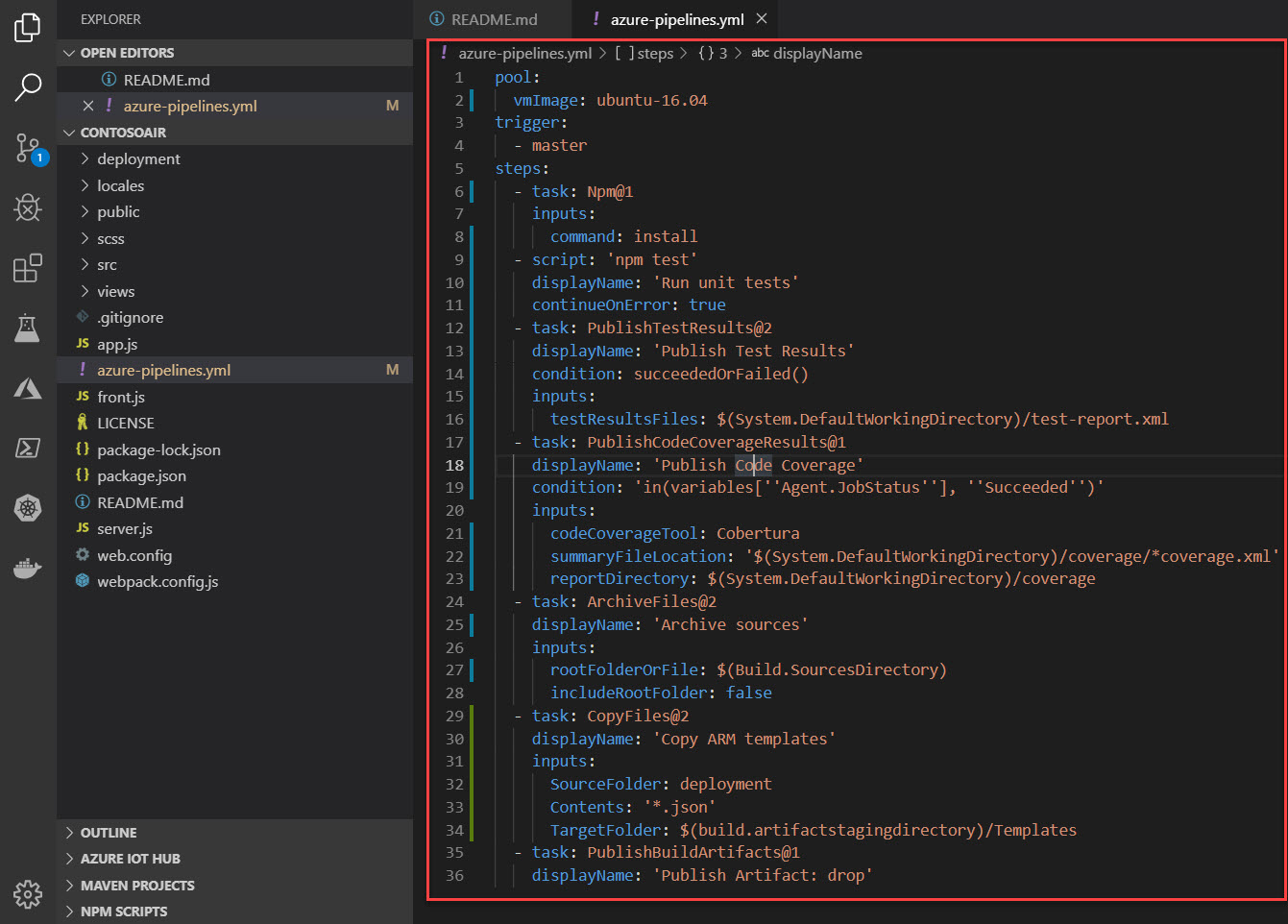
Check README.md file. Badge Available



Select Explorer and Open azure-pipelines.yml



Replace with below code



pool:

vmImage: ubuntu-16.04

trigger:

- master

steps:

- task: Npm@1

inputs:

command: install

- script: 'npm test'

displayName: 'Run unit tests'

continueOnError: true

- task: PublishTestResults@2

displayName: 'Publish Test Results'

condition: succeededOrFailed()

inputs:

testResultsFiles: $(System.DefaultWorkingDirectory)/test-report.xml

- task: PublishCodeCoverageResults@1

displayName: 'Publish Code Coverage'

condition: 'in(variables[''Agent.JobStatus''], ''Succeeded'')'

inputs:

codeCoverageTool: Cobertura

summaryFileLocation: '$(System.DefaultWorkingDirectory)/coverage/\*coverage.xml'

reportDirectory: $(System.DefaultWorkingDirectory)/coverage

- task: ArchiveFiles@2

displayName: 'Archive sources'

inputs:

rootFolderOrFile: $(Build.SourcesDirectory)

includeRootFolder: false

- task: CopyFiles@2

displayName: 'Copy ARM templates'

inputs:

SourceFolder: deployment

Contents: '\*.json'

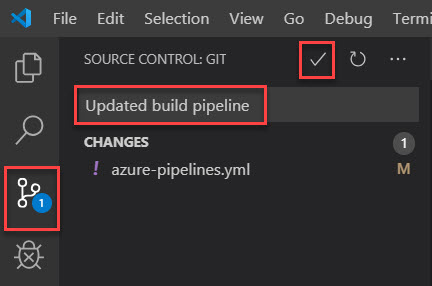
TargetFolder: $(build.artifactstagingdirectory)/Templates

- task: PublishBuildArtifacts@1

displayName: 'Publish Artifact: drop'

Click on Source Control and write message: **Updated build pipeline**.

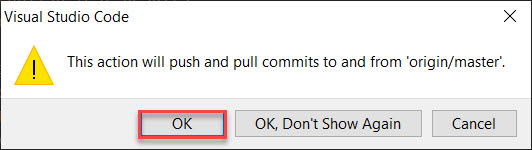
Click on Tick icon



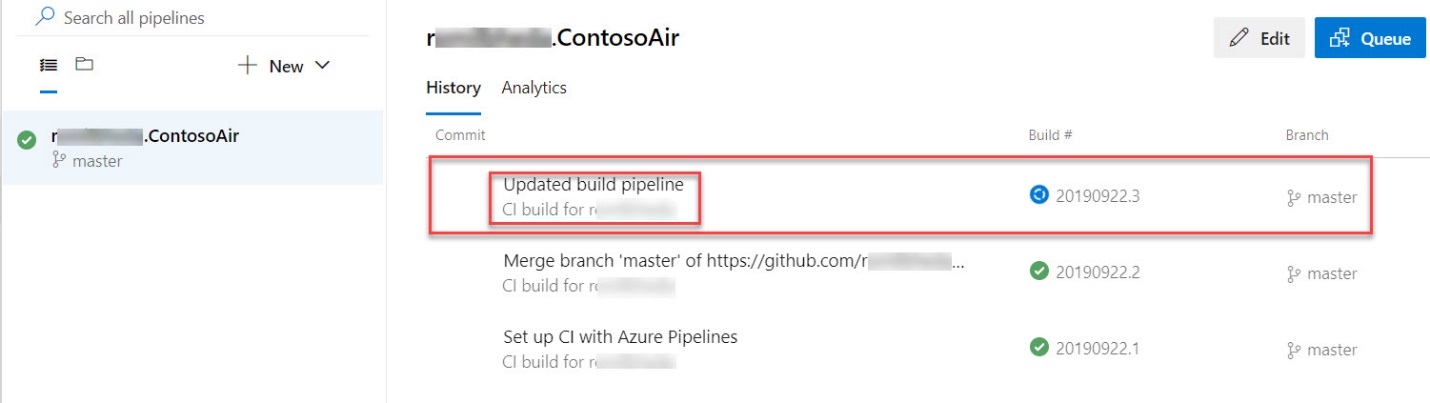
Once again click on Chnages.



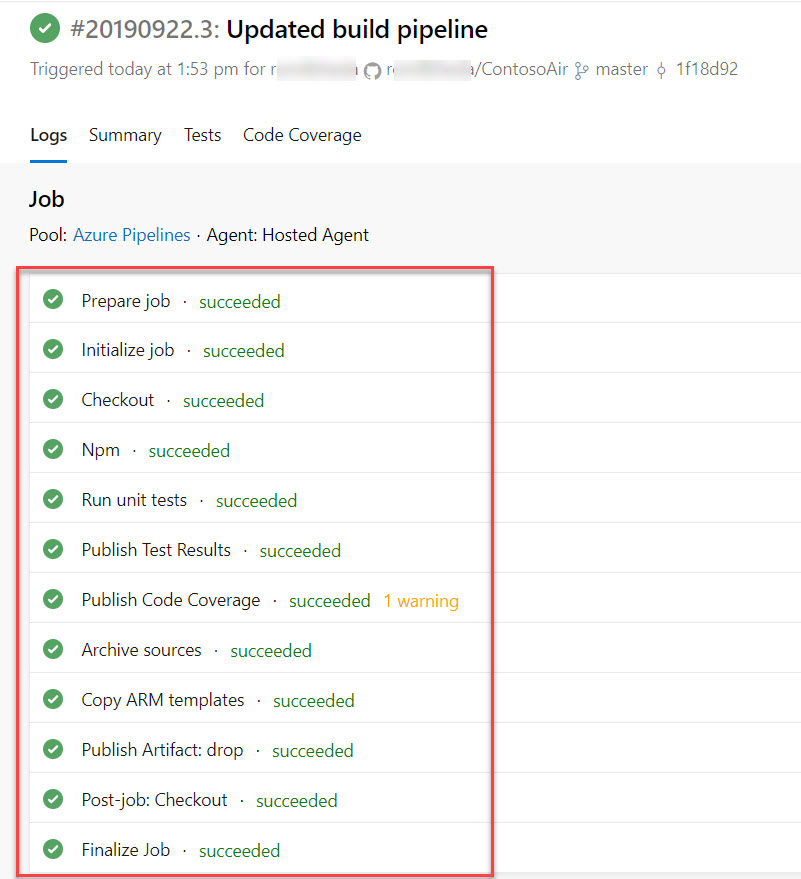
Click on OK button



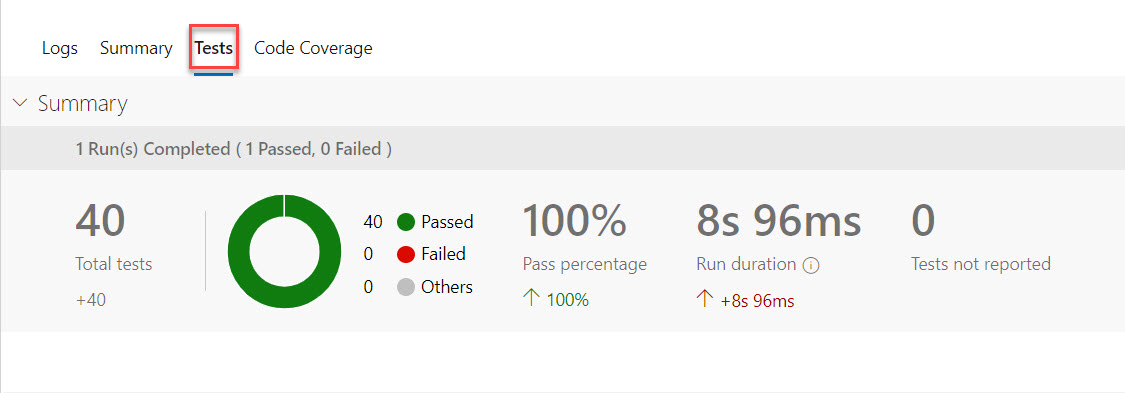
Navigate to Azure DevOps portal and check pipeline – Updated build pipeline



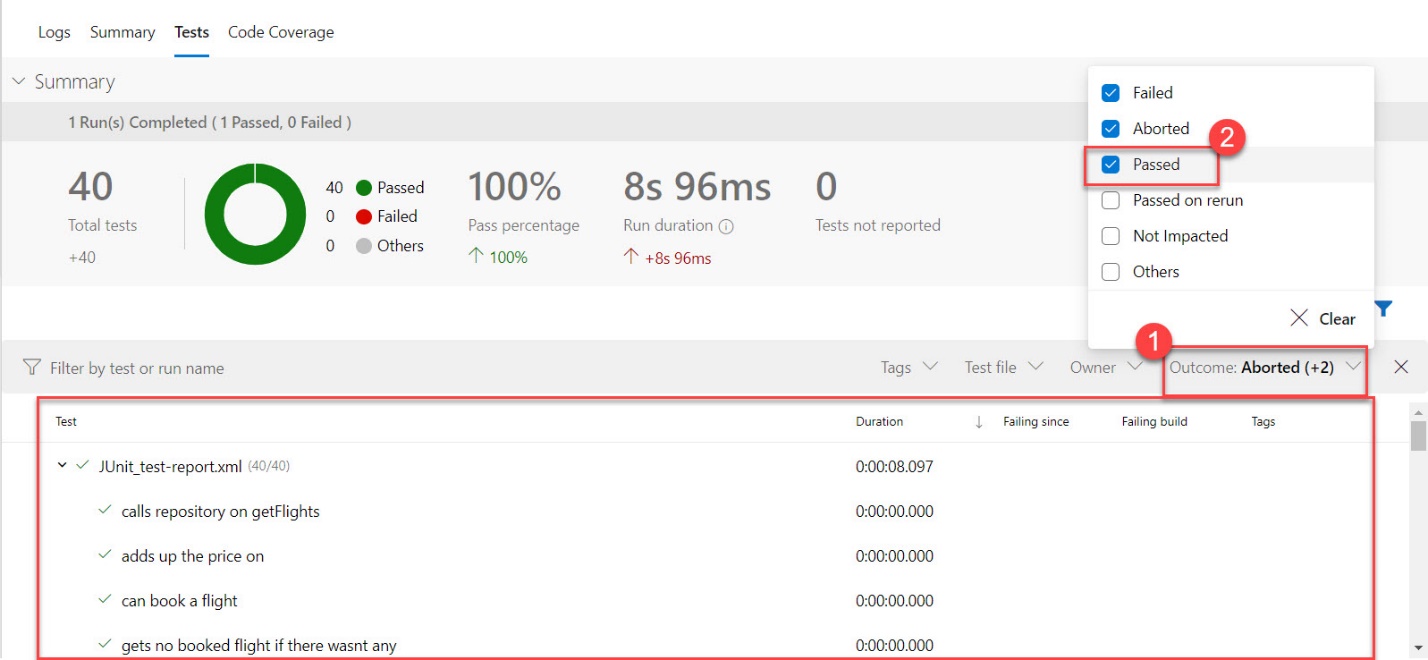
Wait for few minutes to complete job



Click on Tests



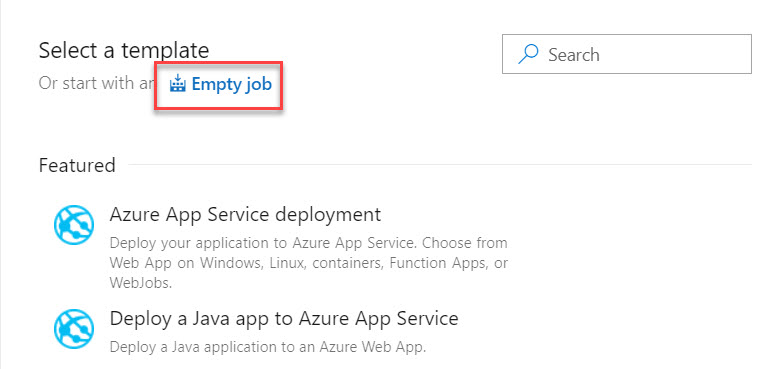
Click on Outcome and add Passed option. Result available

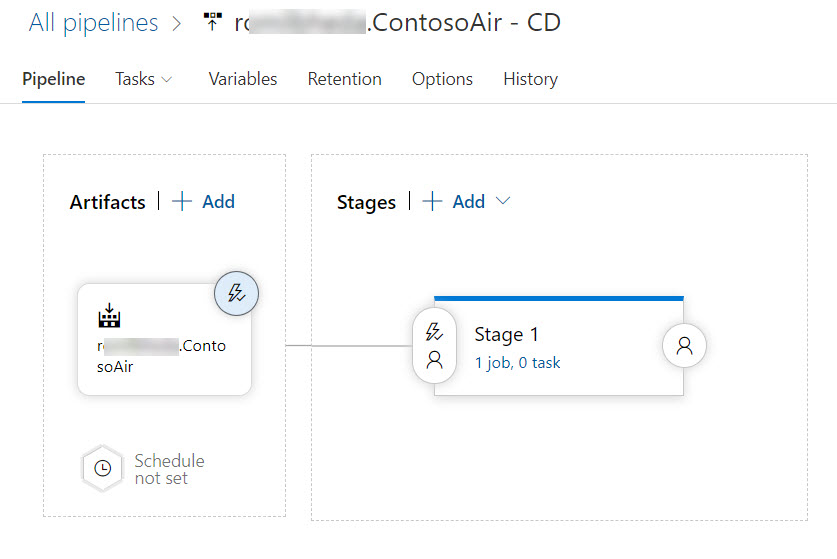


Click on Release



Click on **Empty Job**



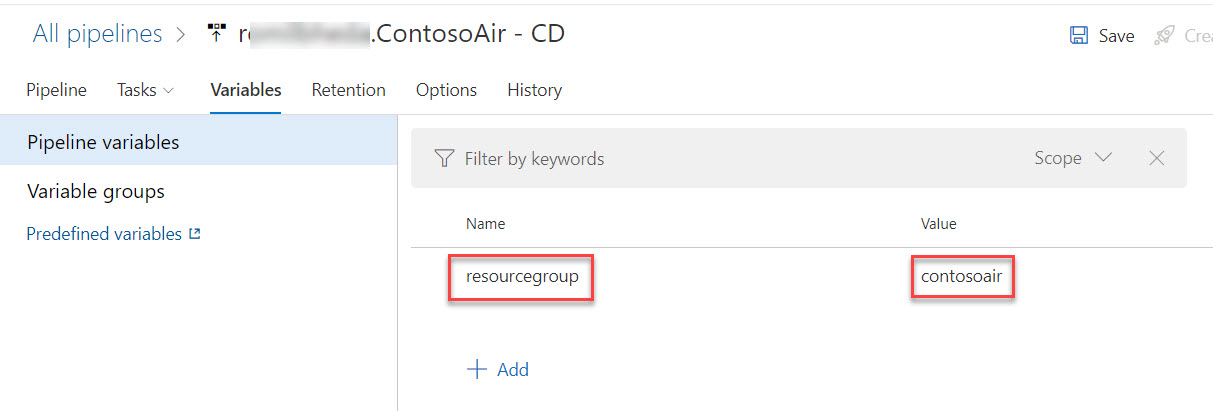


Click on variables -> + Add option



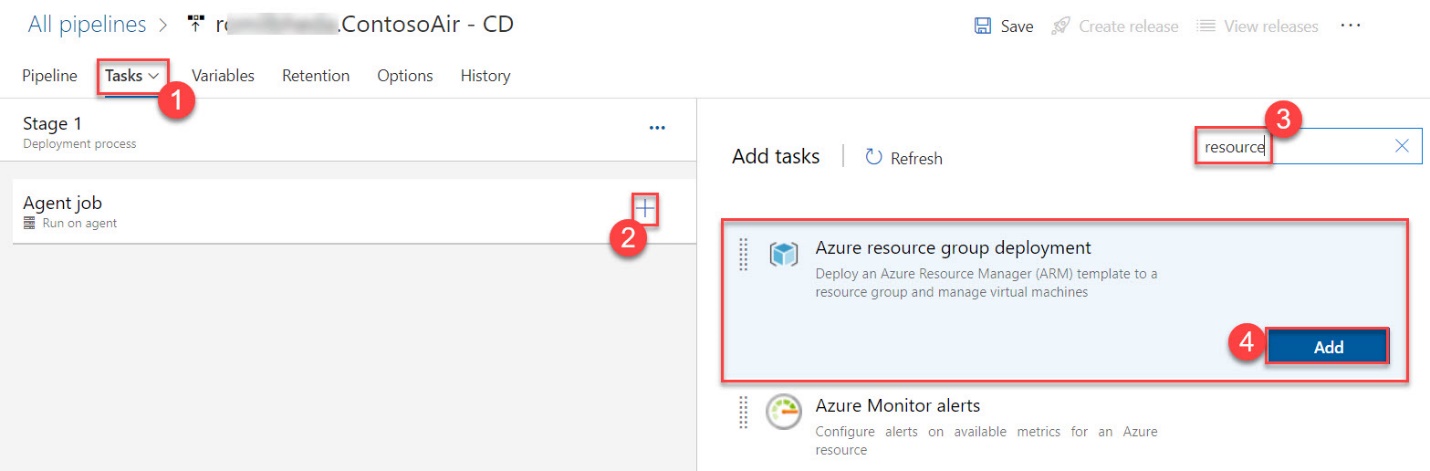
Name: resourcegroup

Value: contosoair

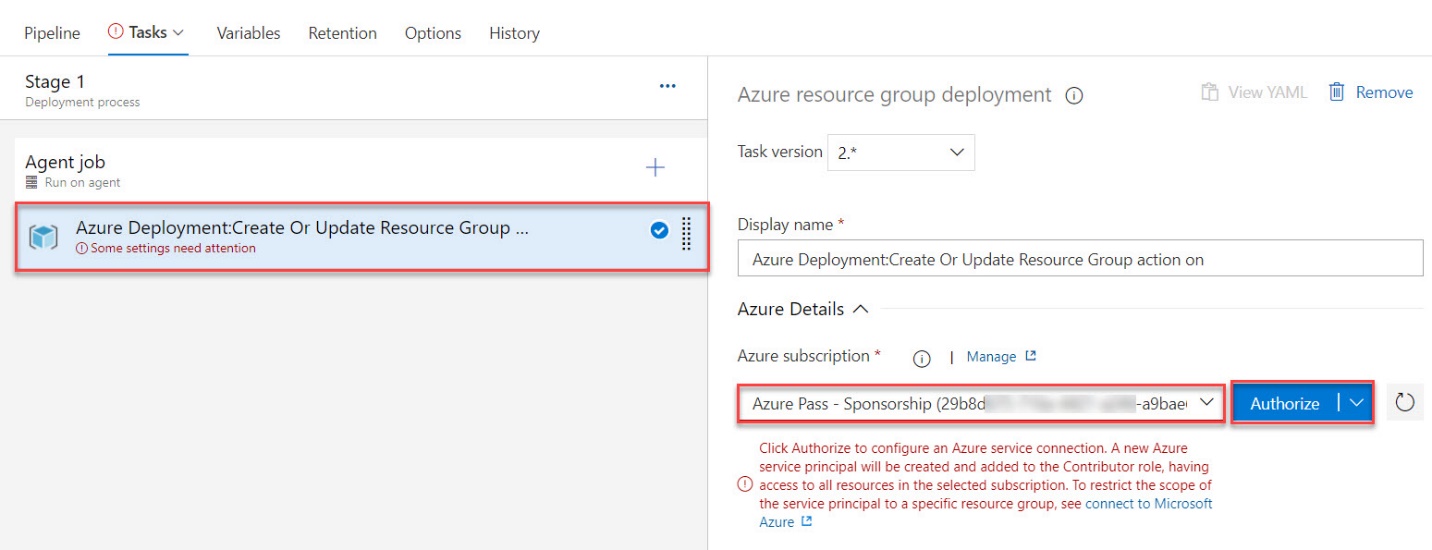


Click on Tasks -> + Add -> search for “resource”

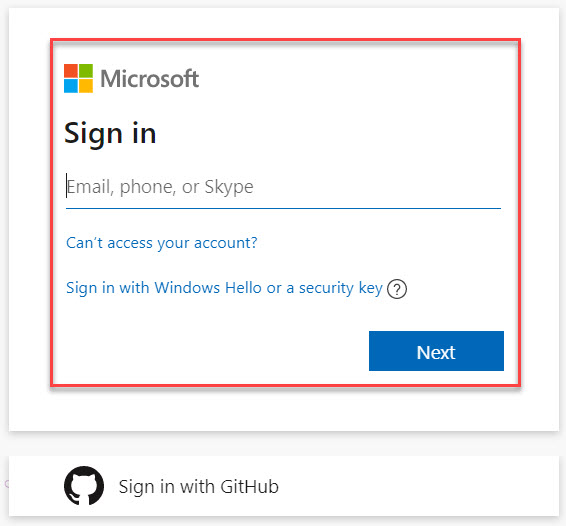
Select Azure resource group deployment and click on Add button



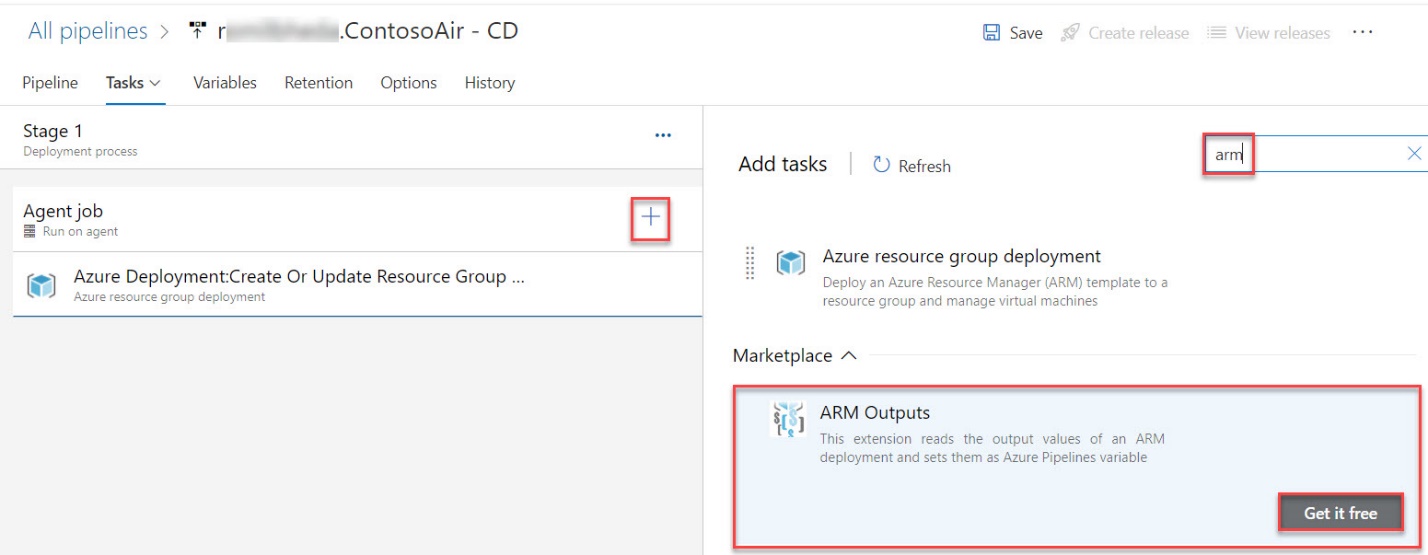
Select your subscription



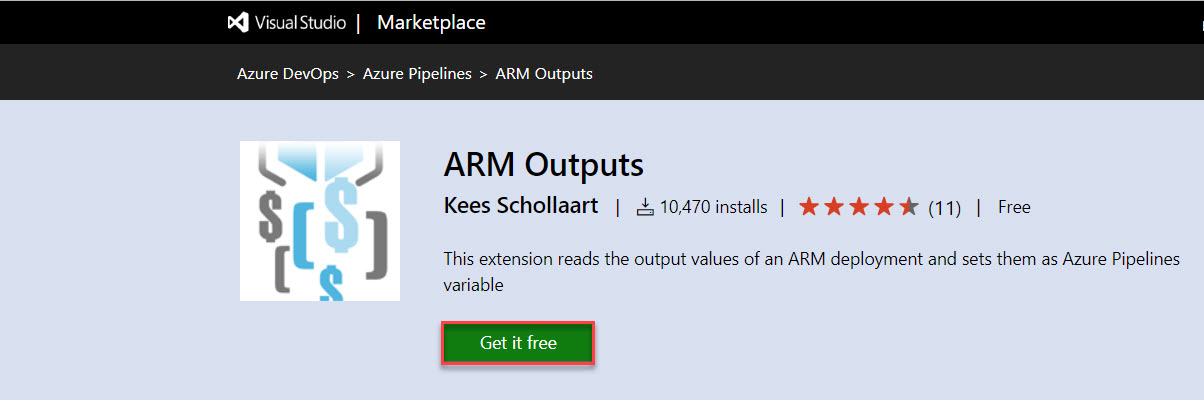
Enter Credentials



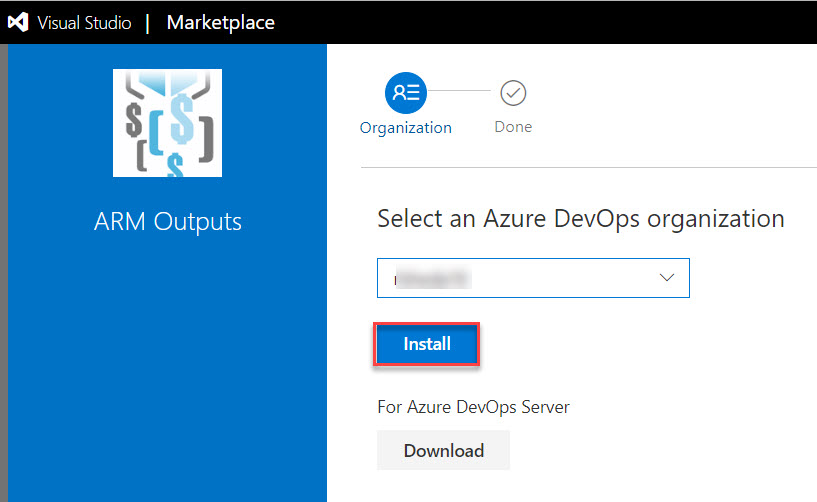
Click on **+** and add search for **arm**



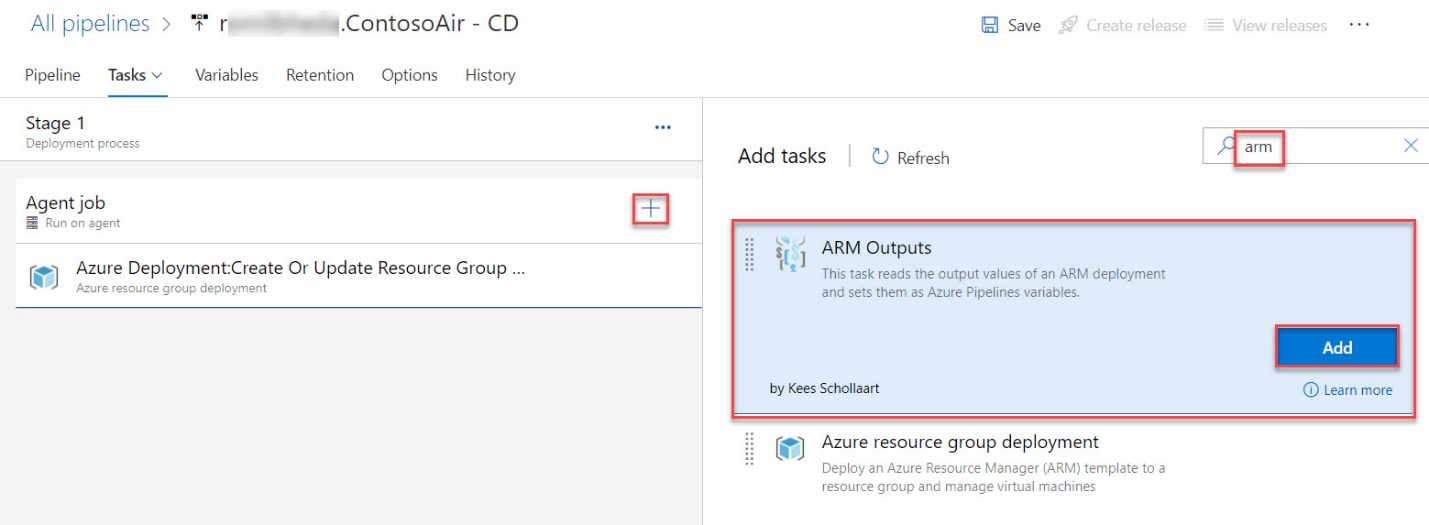
Click on Get it free



Install

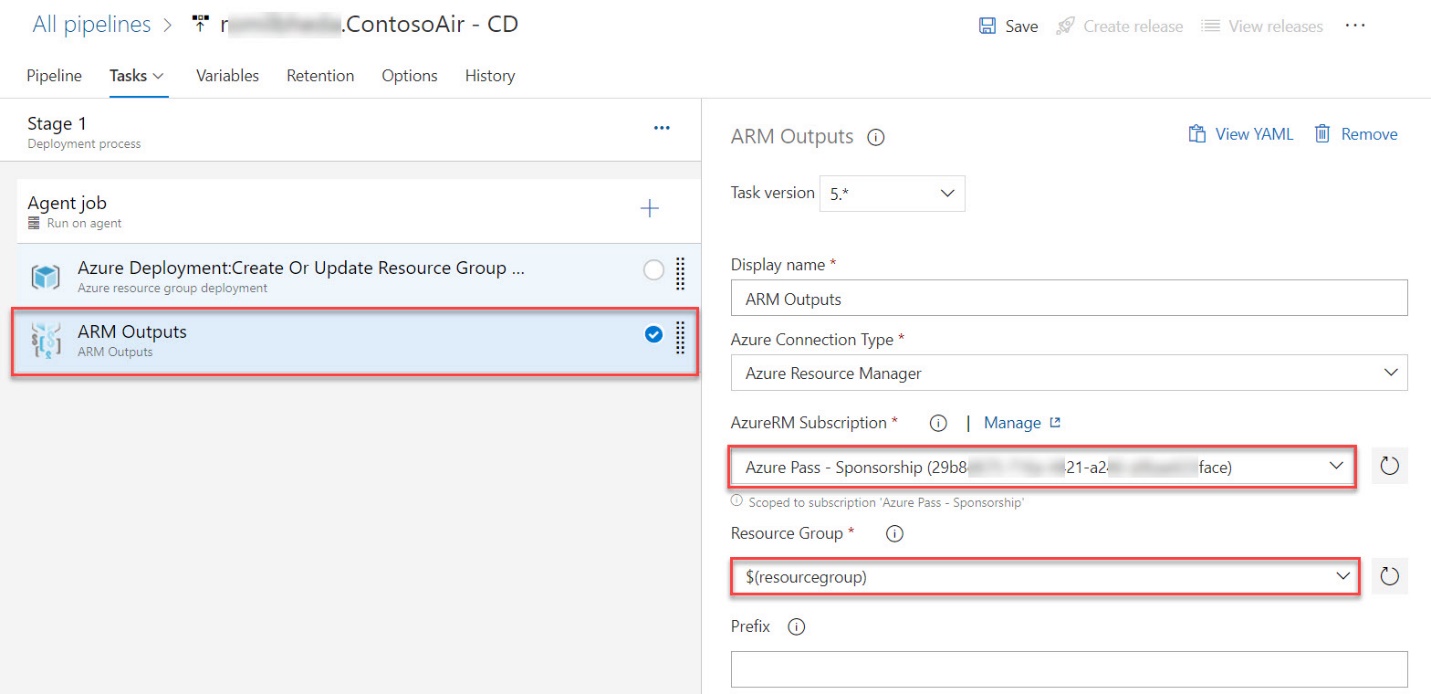


Back to your ARM Output and click on Add option



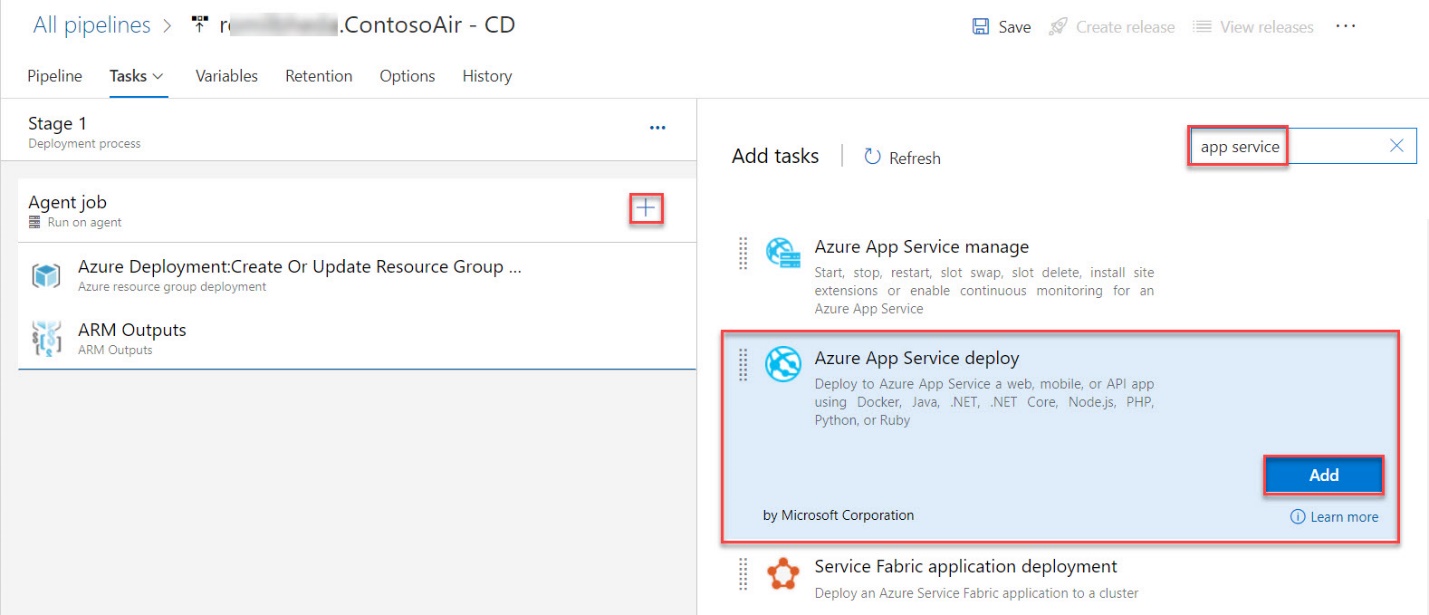
Select Azure Subscription

Resource Group: **$(resourcegroup)**



Add on more job. Search for app service

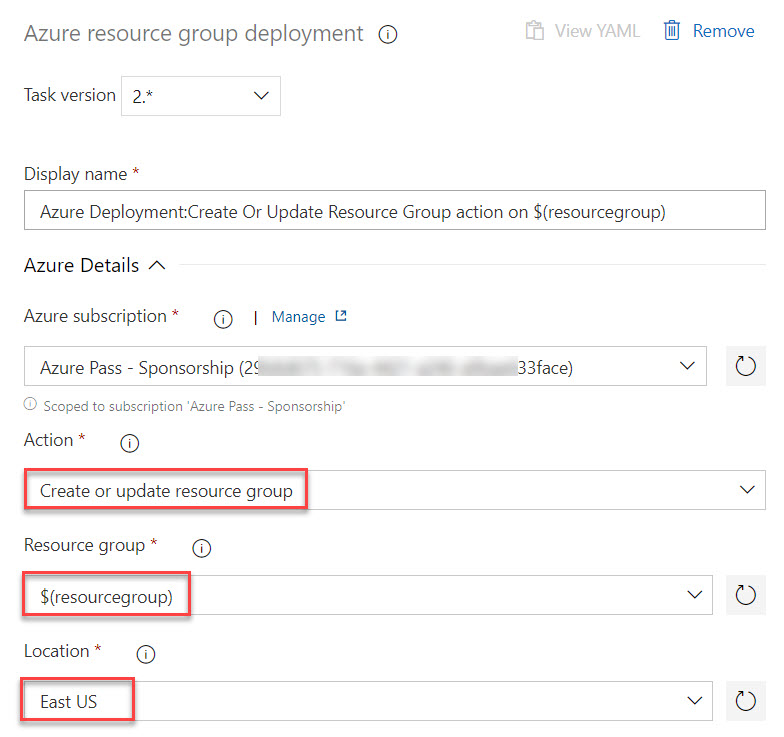
Click on Add



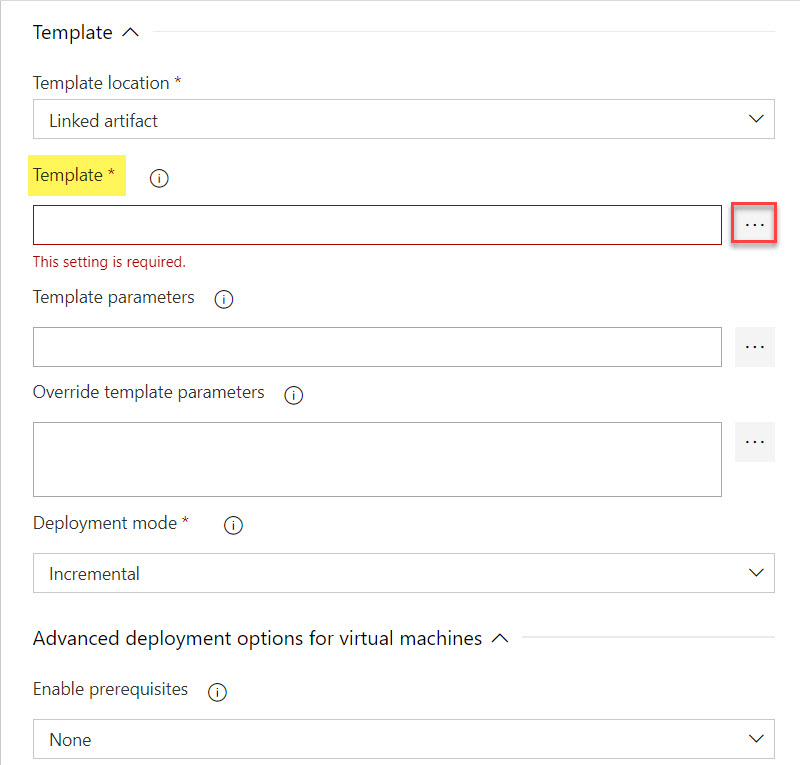
Action: **Create or update resource group**

Resource group: **$(resourcegroup)**

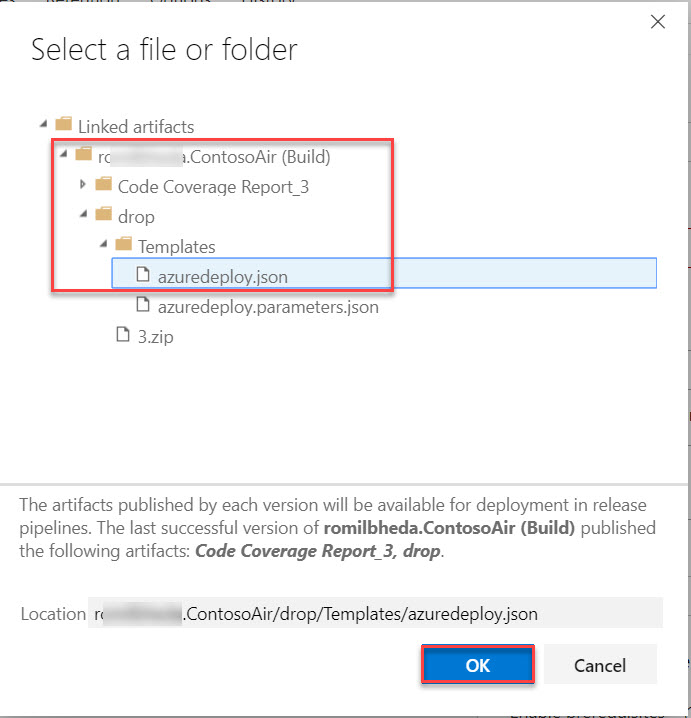
Location: **Any location**



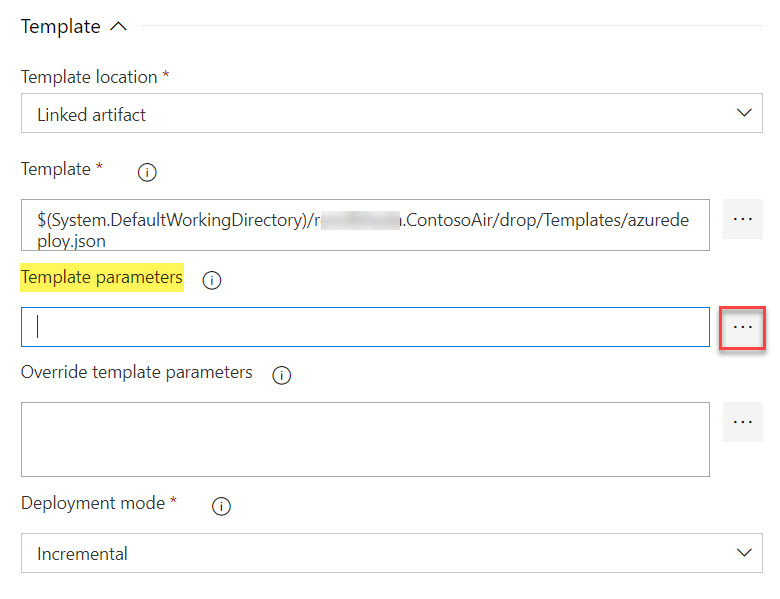
Template – Select Browse option



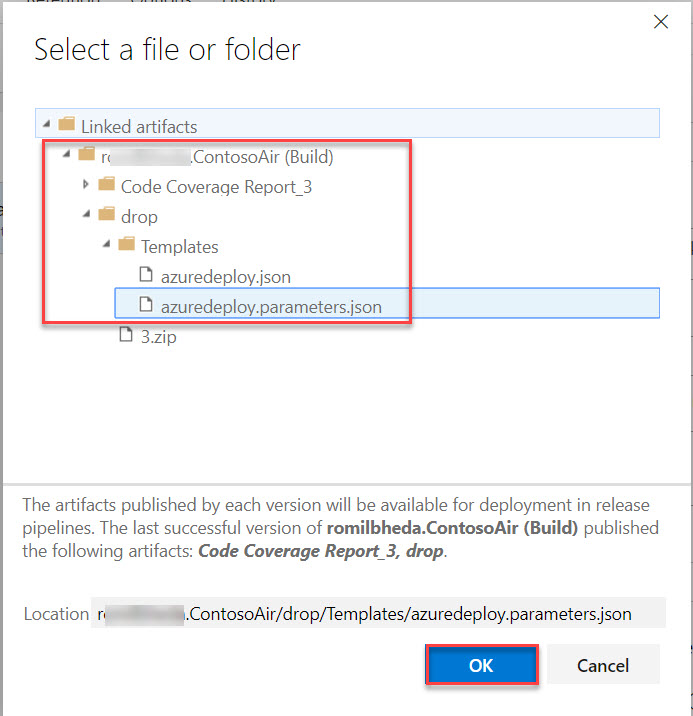
Expand and select **azuredeploy.json** file



Template Parameters – Browse option

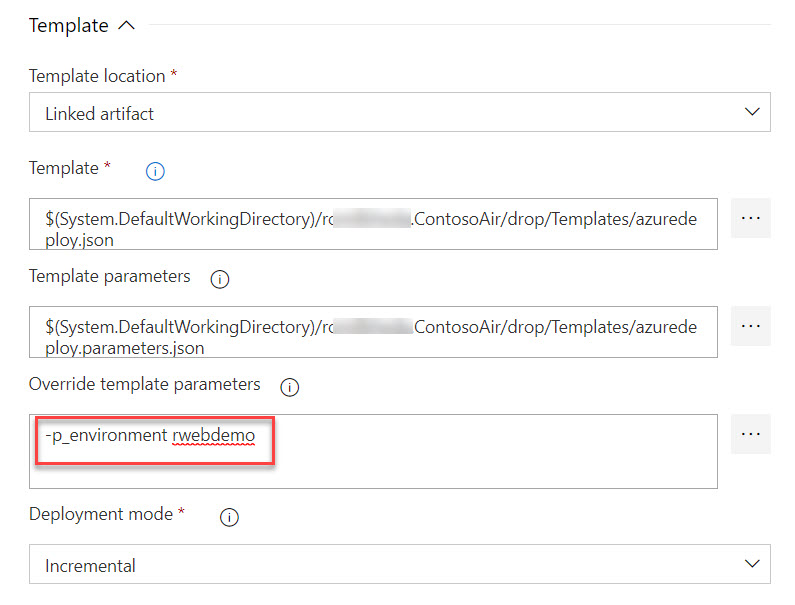


Expand and select **azuredeploy.parameters.json** file



Add template parameters (App unique name)

**-p\_environment rwebdemo**

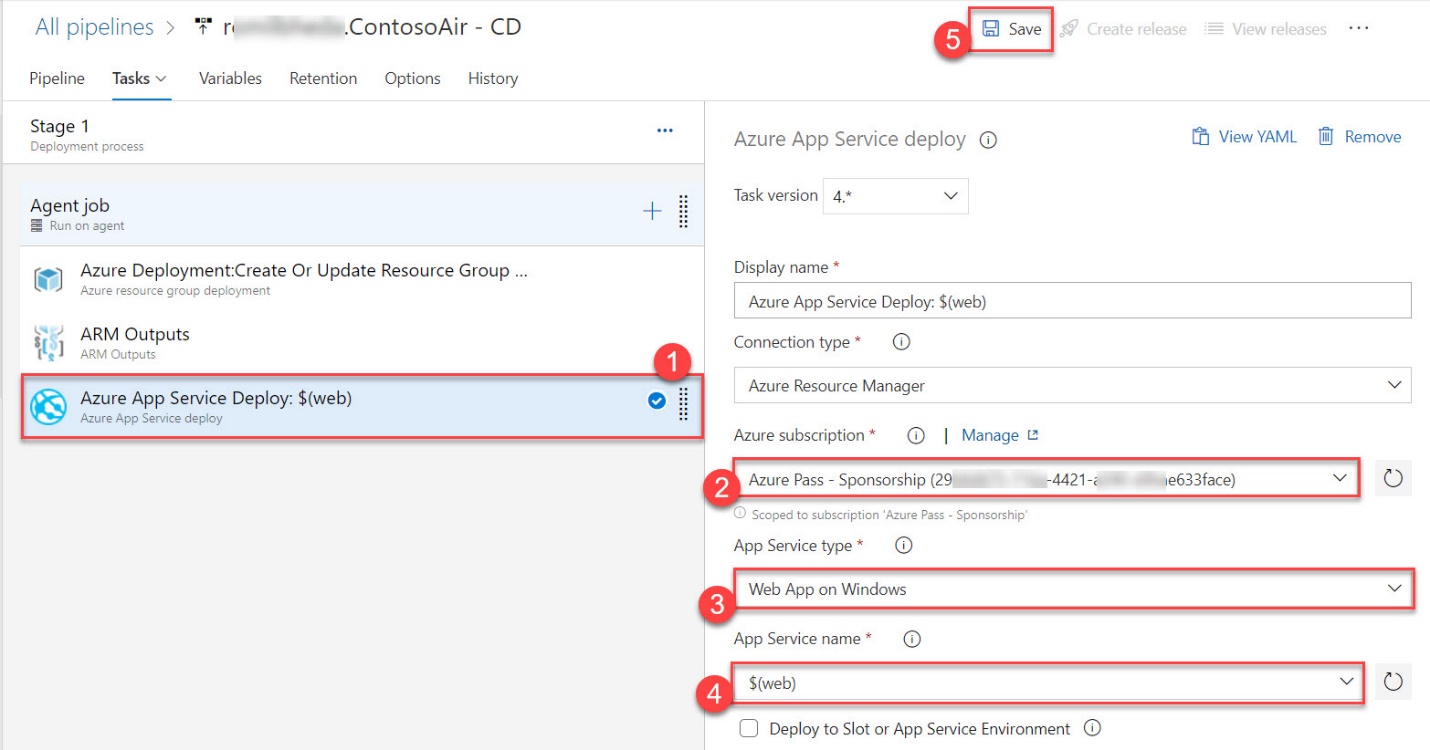


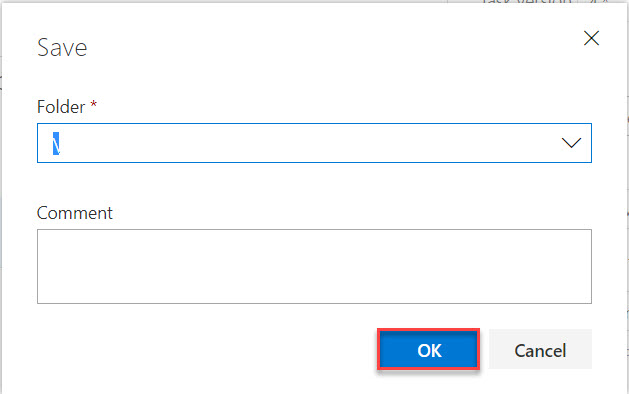
Select **Azure App Service Deploy**

Select **Subscription**

App Service type: **Web App on Windows**

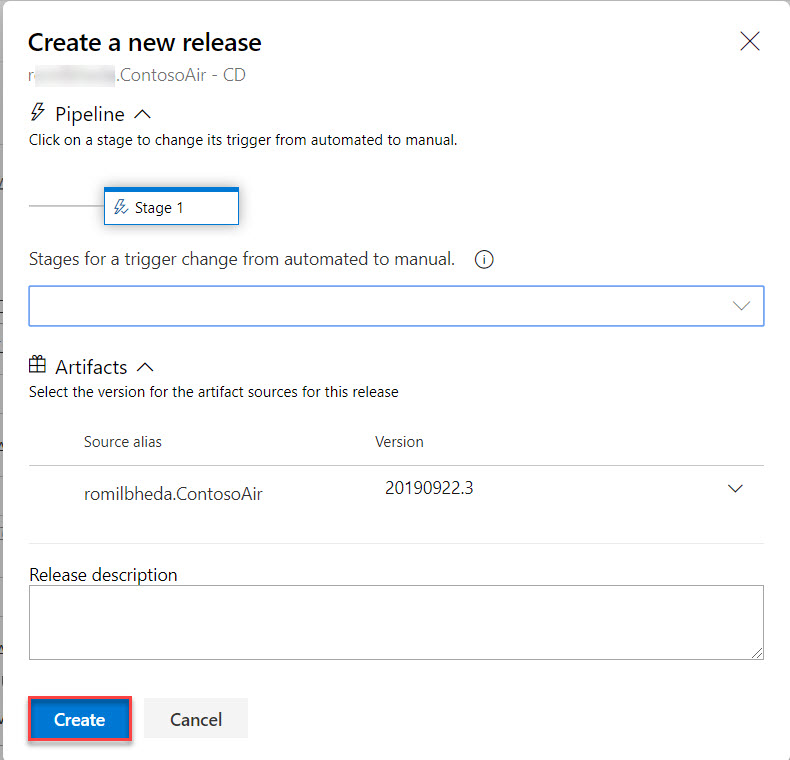
App Service Name: **$(web)**



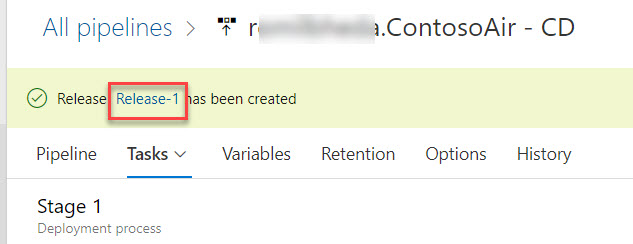


Click on Create release option

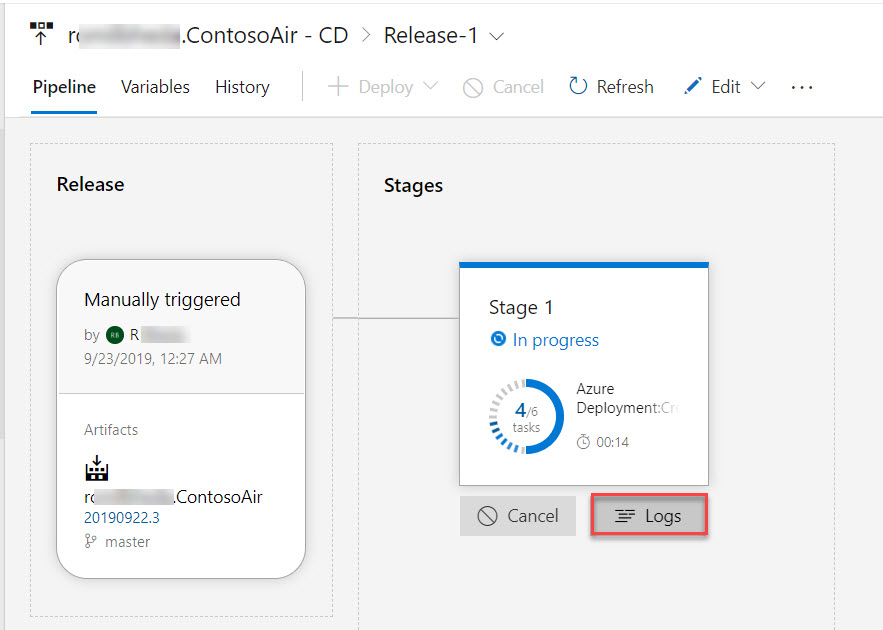




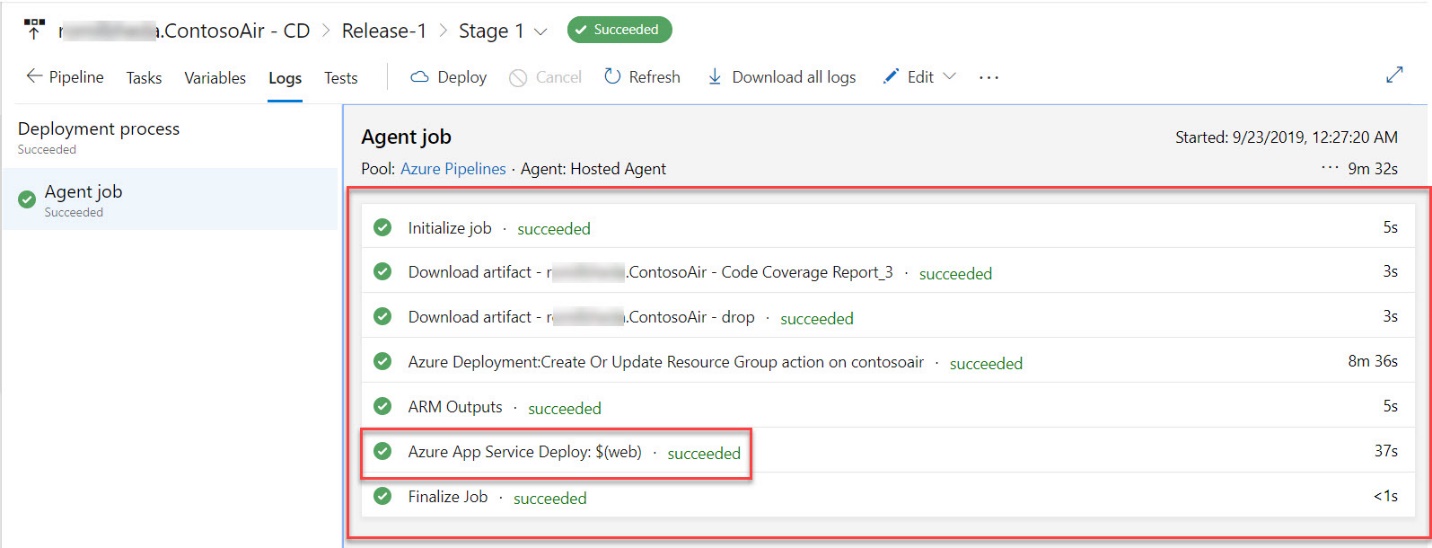
Click on Release-x option



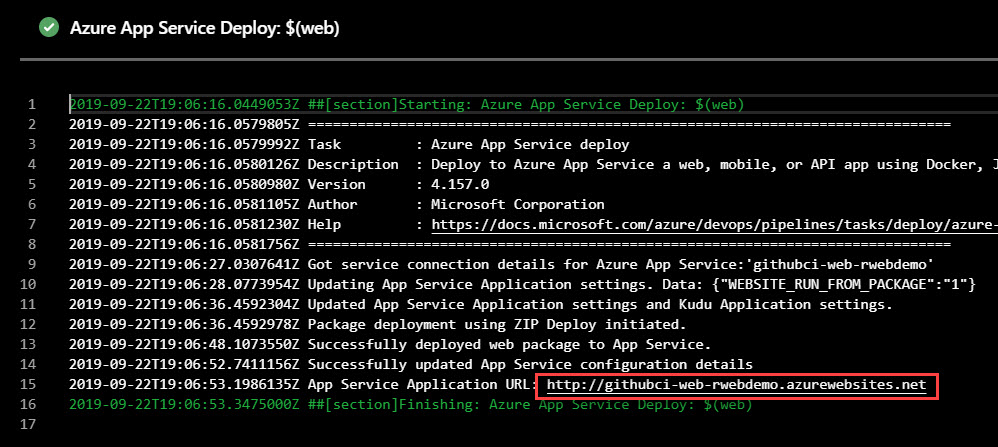
Logs



Click on Azure App Service Deploy: $(web) task



Click on App Service Application URL



Project is running

