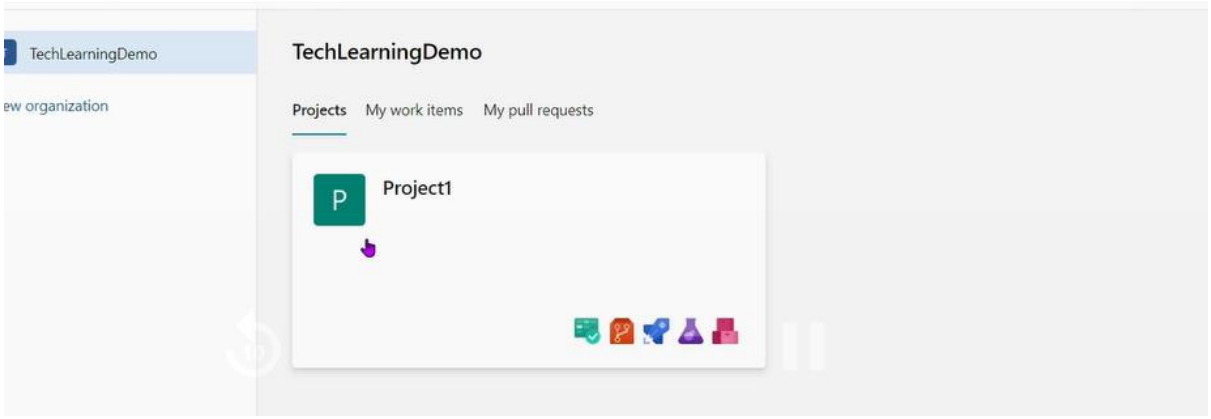
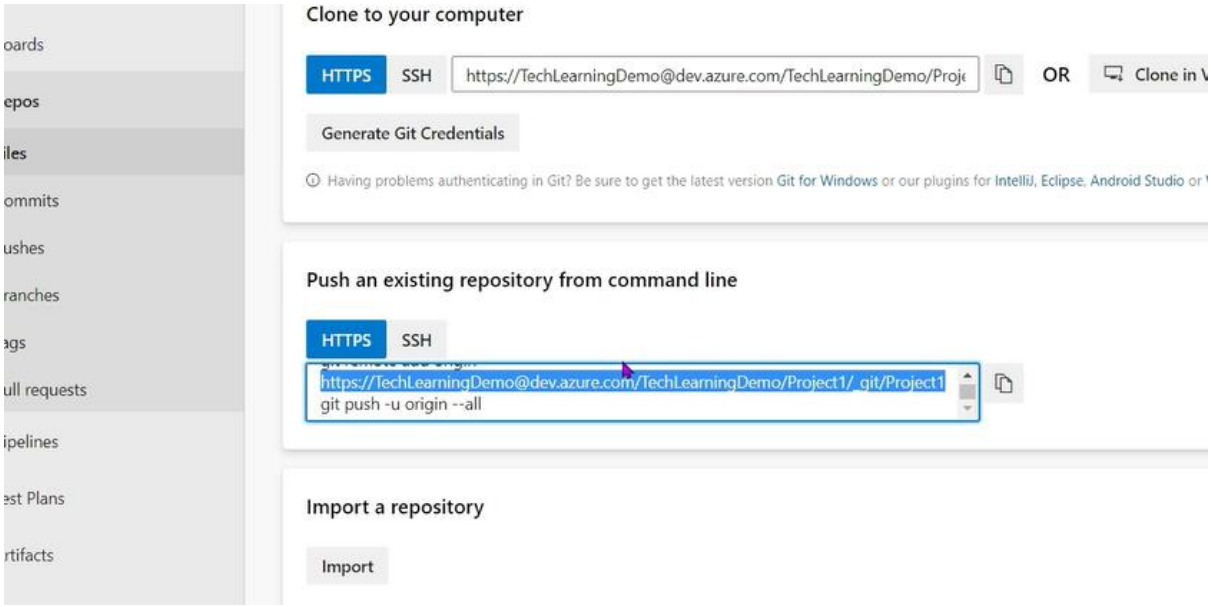
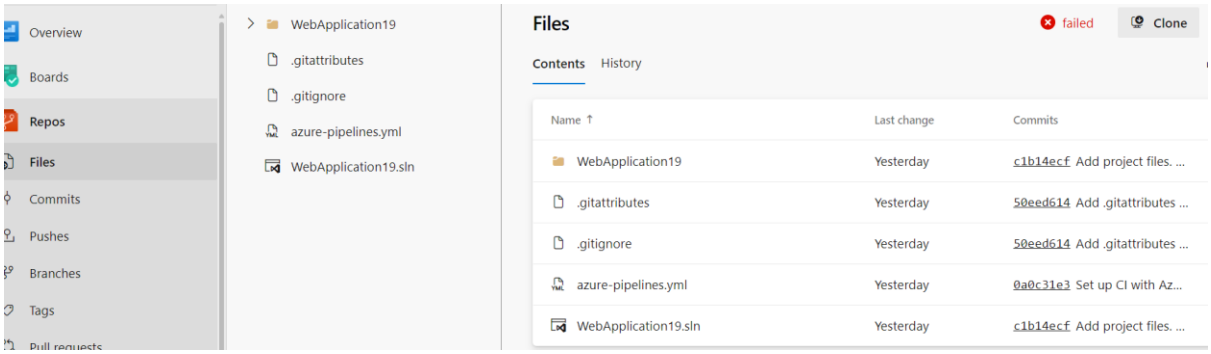
ARM Templates Integration with Azure CI/CD Pipelines

Create a Project on Azure DevOps Portal

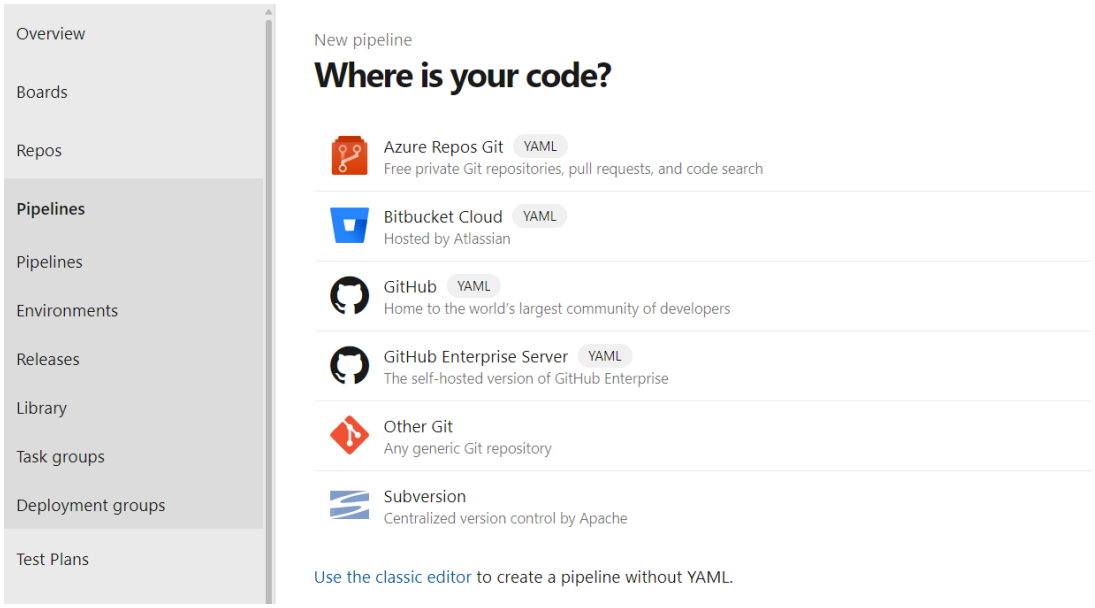




Create ASP.Net web Application in Visual Studio and push it to Azure Repos.



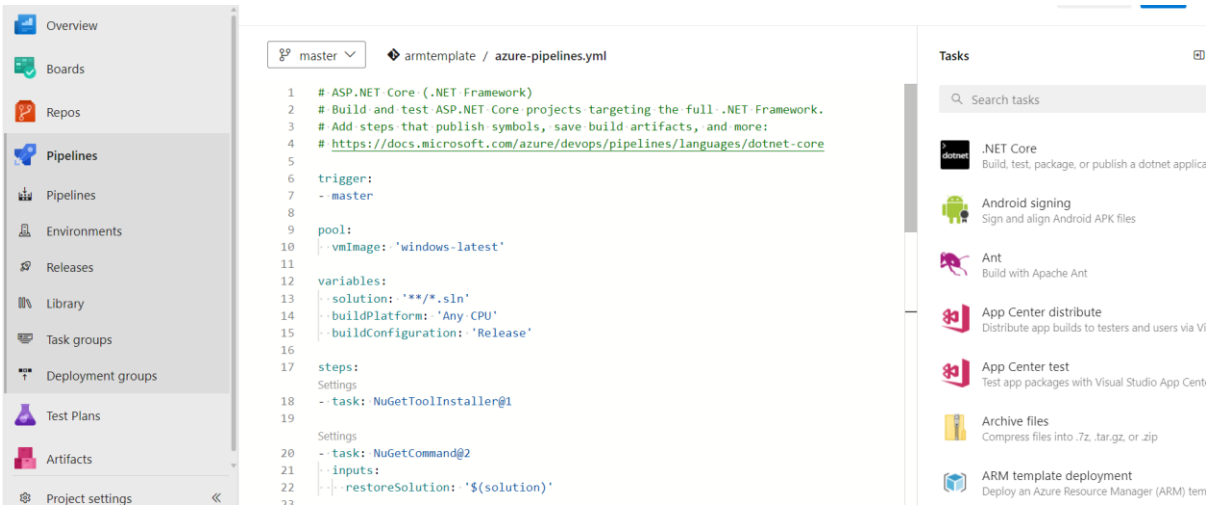
Create Build Pipeline and select it from Azure repos Git and select ASP.Net Framework

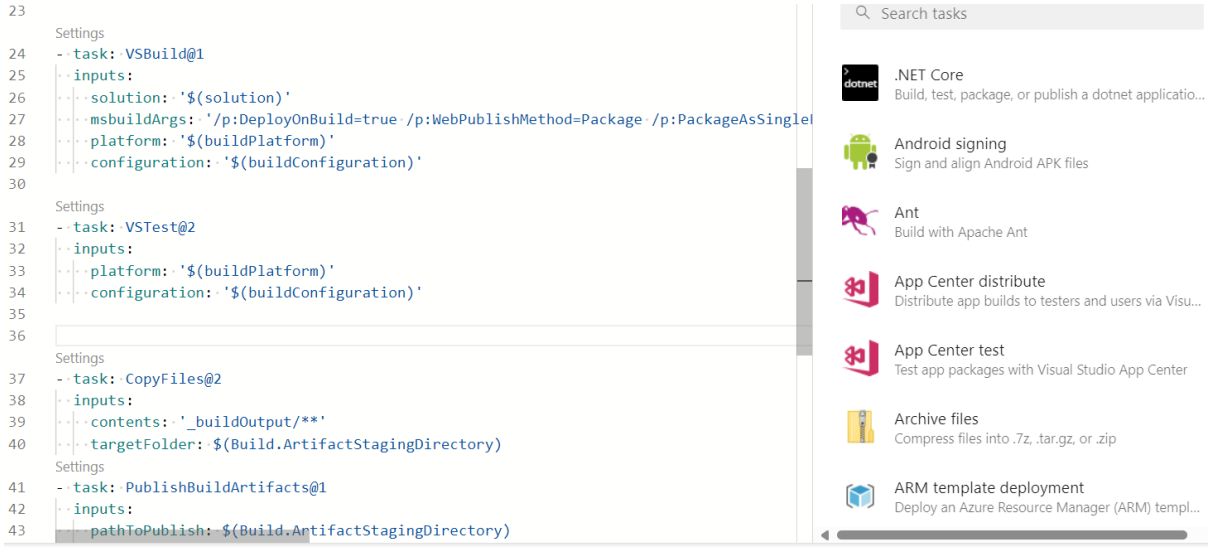


YAML Pipeline Code to build the solution and Publish Artifacts.

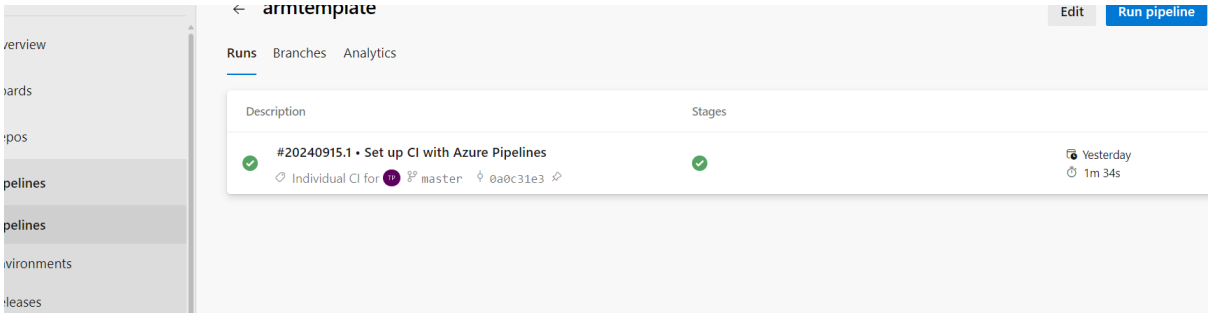
```
# ASP.NET Core (.NET Framework)
# Build and test ASP.NET Core projects targeting the full .NET Framework.
# Add steps that publish symbols, save build artifacts, and more:
# https://docs.microsoft.com/azure/devops/pipelines/languages/dotnet-core

trigger:
- master

pool:
  vmImage: 'windows-latest'

variables:
  solution: '**/*.sln'
  buildPlatform: 'Any CPU'
  buildConfiguration: 'Release'

steps:
- task: NuGetToolInstaller@1

- task: NuGetCommand@2
  inputs:
    restoreSolution: '$(solution)'

- task: VSBuild@1
  inputs:
    solution: '$(solution)'
```

```
    msbuildArgs: '/p:DeployOnBuild=true /p:WebPublishMethod=Package /p:PackageAsS
ingleFile=true /p:SkipInvalidConfigurations=true /p:DesktopBuildPackageLocation="$(b
uild.artifactStagingDirectory)\WebApp.zip" /p:DeployIisAppPath="Default Web Site"'
    platform: '$(buildPlatform)'
    configuration: '$(buildConfiguration)'

- task: VSTest@2
  inputs:
    platform: '$(buildPlatform)'
    configuration: '$(buildConfiguration)'

- task: CopyFiles@2
  inputs:
    contents: '_buildOutput/**'
    targetFolder: $(Build.ArtifactStagingDirectory)
- task: PublishBuildArtifacts@1
  inputs:
    pathToPublish: $(Build.ArtifactStagingDirectory)
    artifactName: MyBuildOutputs
```

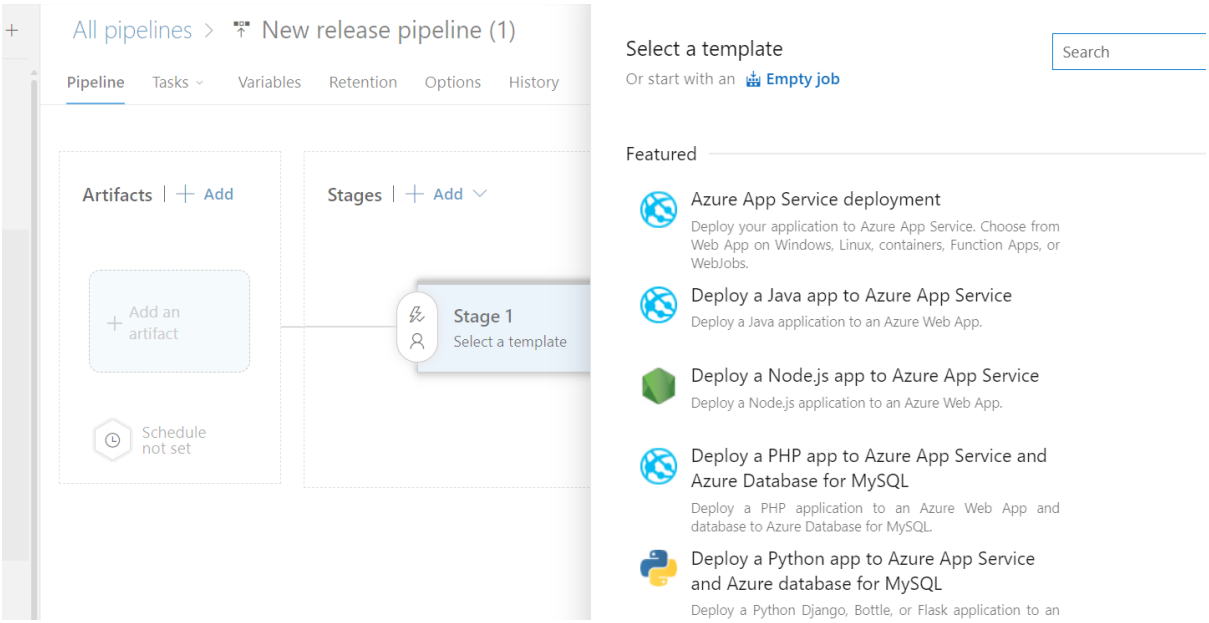




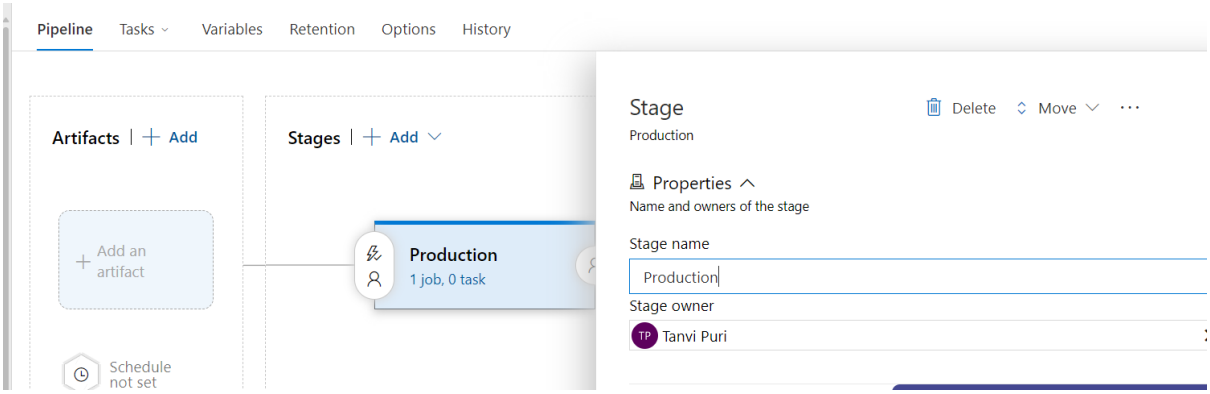
Run the pipeline. It will be success.

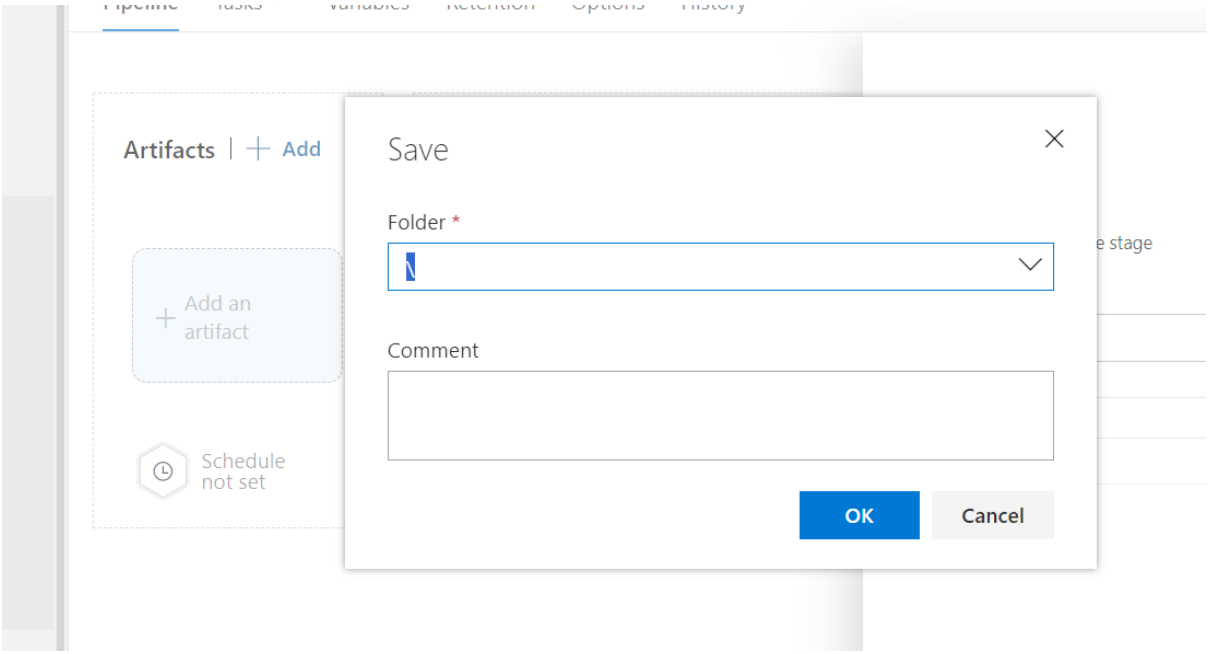


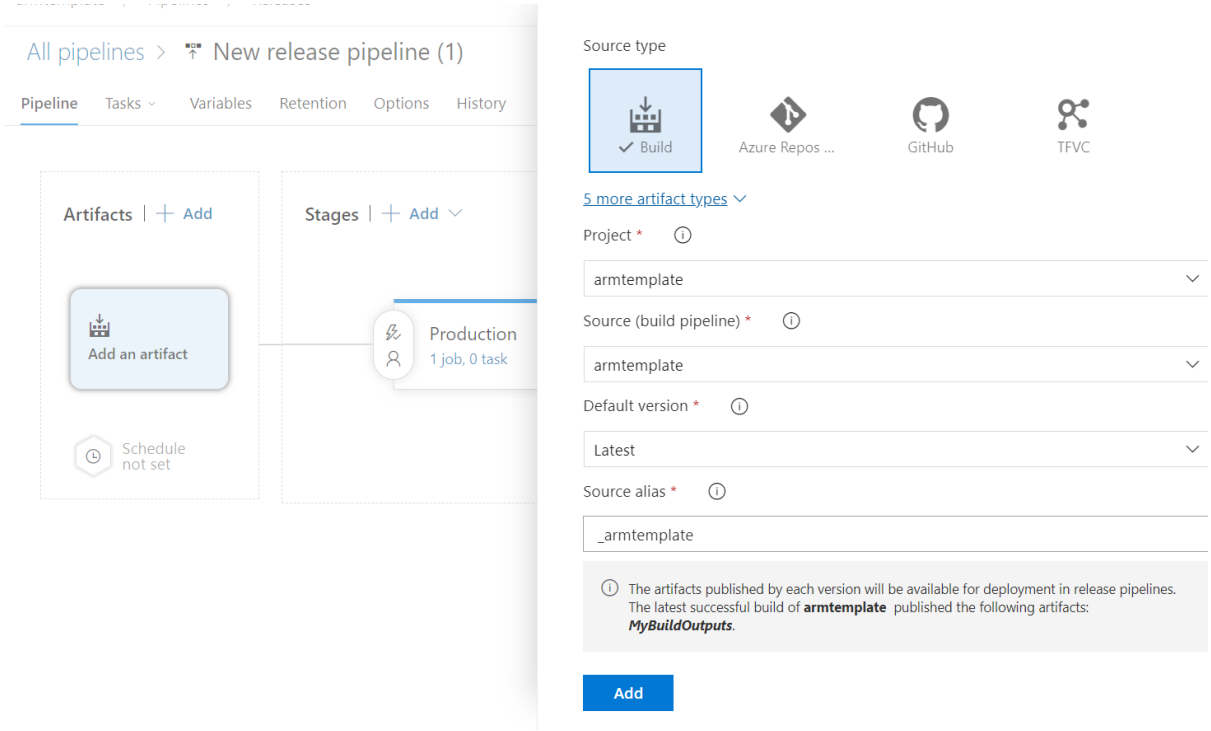
Create new Release Pipeline and select Empty job



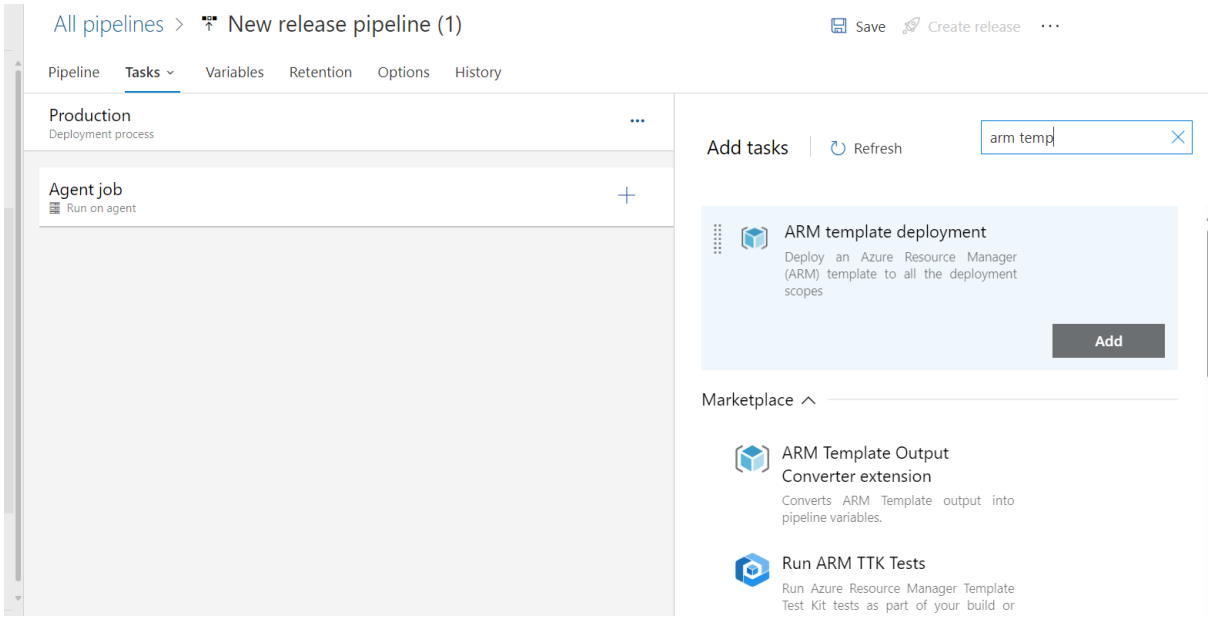
Specify the stage name to Production. Click on Save. Also add to artifacts.



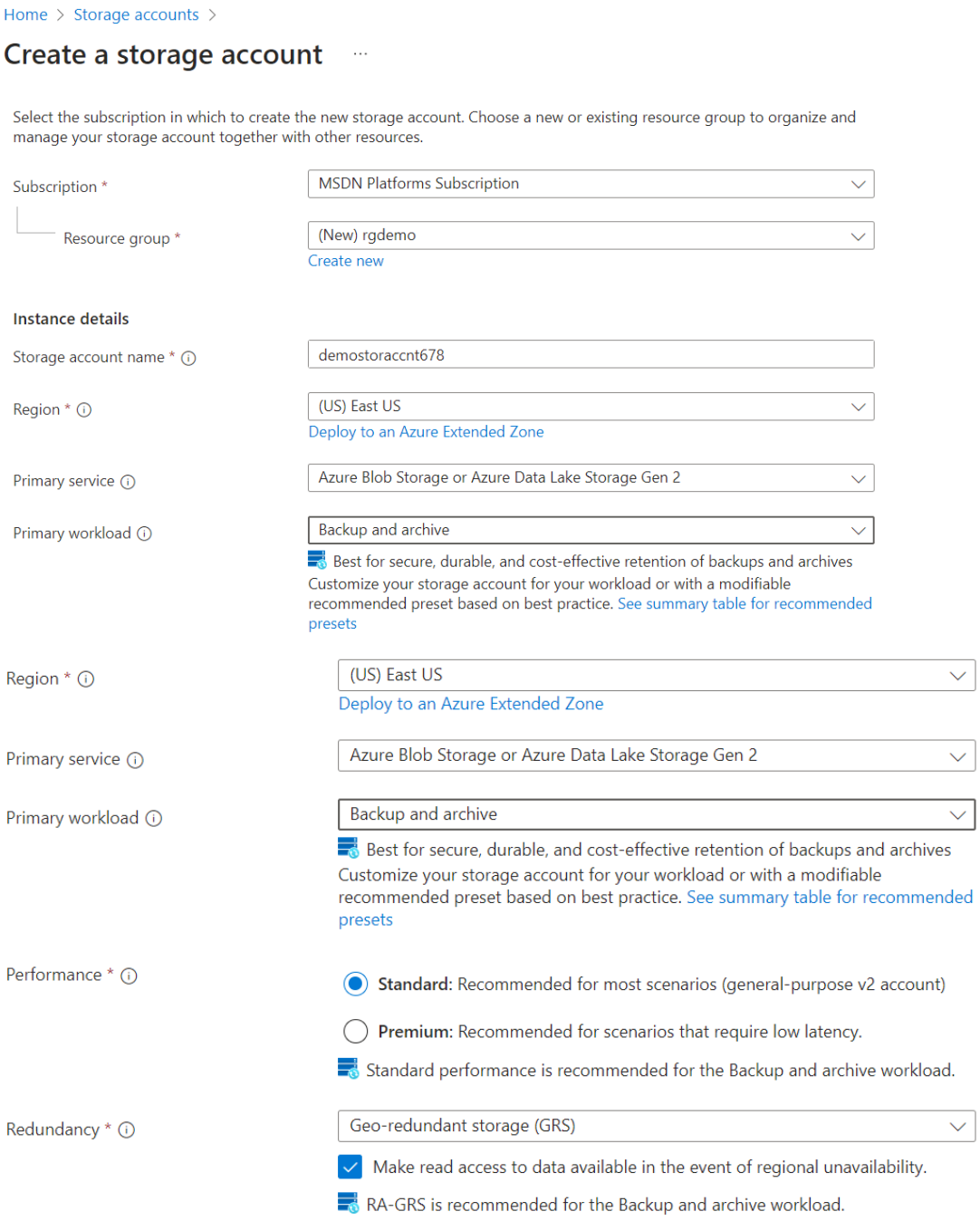




Add the task for ARM Template Deployment



On Azure Portal create storage account as per configurations given in the below screenshot.



Enable Anonymous access on Container. Keep other settings as default.

Basics | **Advanced** | Networking | Data protection | Encryption | Tags | Review + create

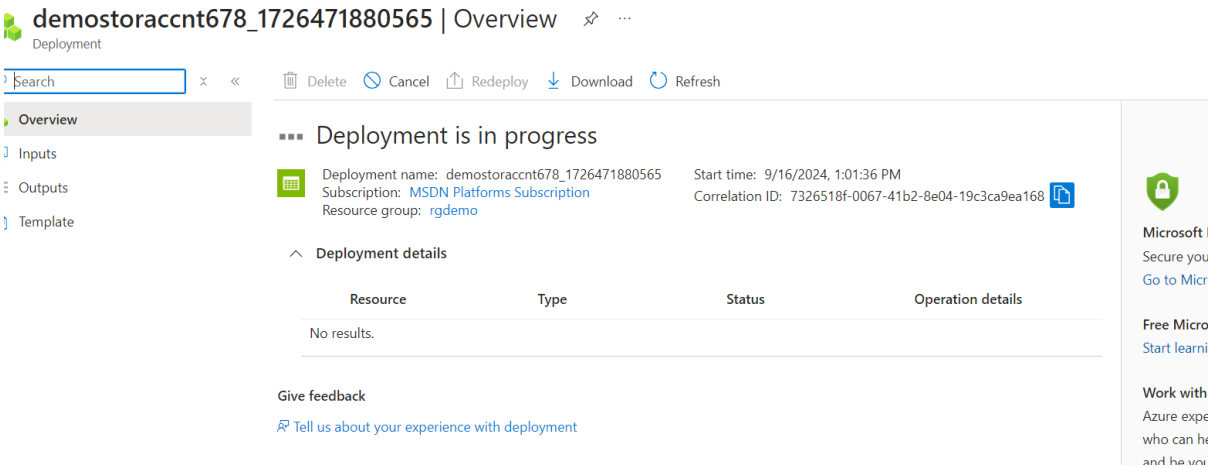
Security

Configure security settings that impact your storage account.

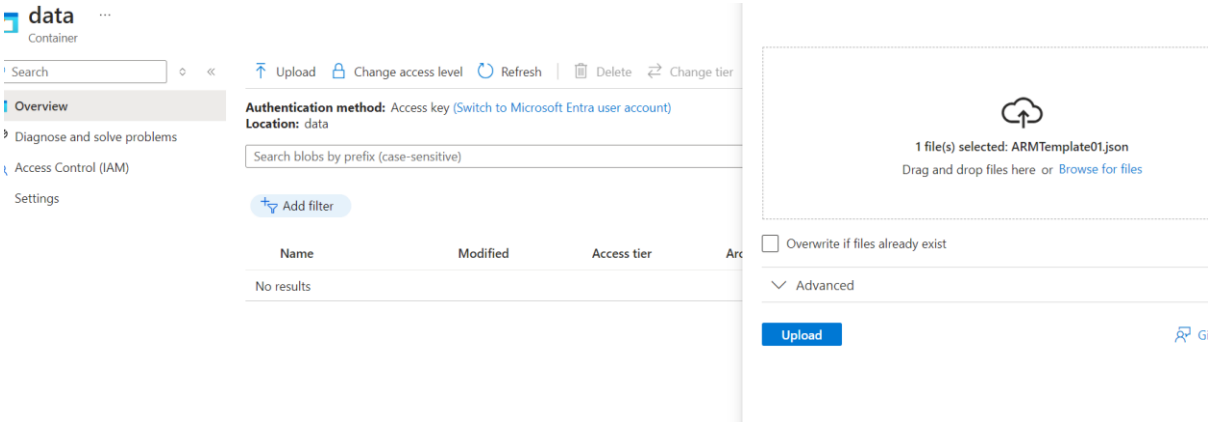
- Require secure transfer for REST API operations ⓘ — ☑
- Allow enabling anonymous access on individual containers ⓘ — ☑
- Enable storage account key access ⓘ — ☑
- Default to Microsoft Entra authorization in the Azure portal ⓘ — ☐
- Minimum TLS version ⓘ — Version 1.2
- Permitted scope for copy operations (preview) ⓘ — From any storage account

Hierarchical Namespace

Hierarchical namespace, complemented by Data Lake Storage Gen2 endpoint, enables file and directory semantics, accelerates big data analytics workloads, and enables access control lists (ACLs) Learn more



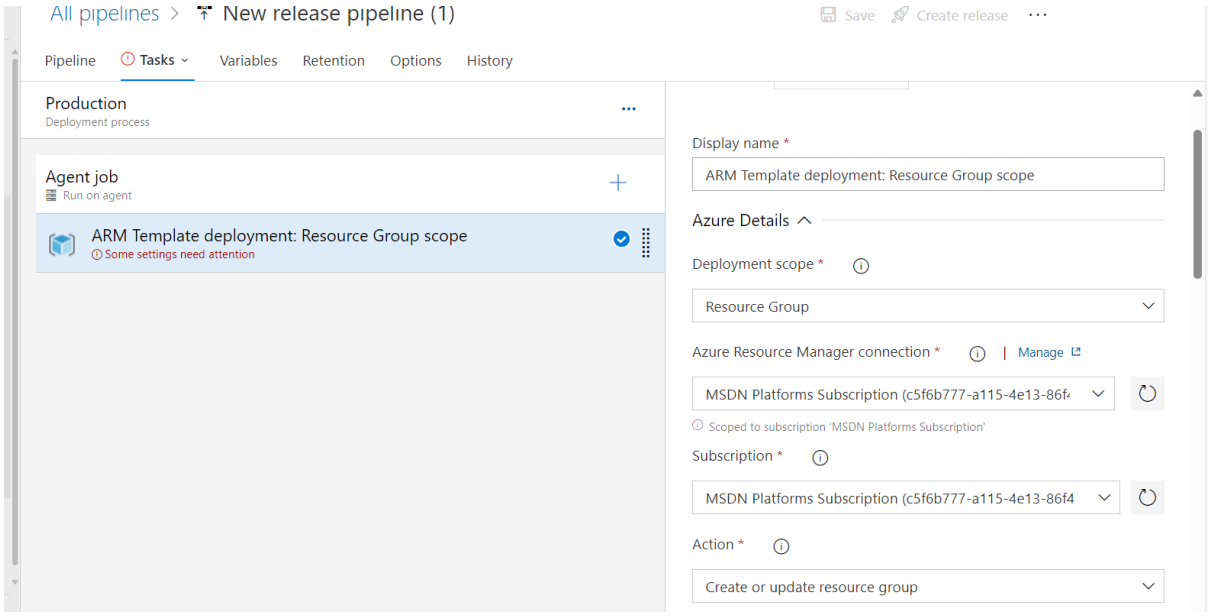
Create a new container, provide read access to container level and upload ARM Template json script to blob.



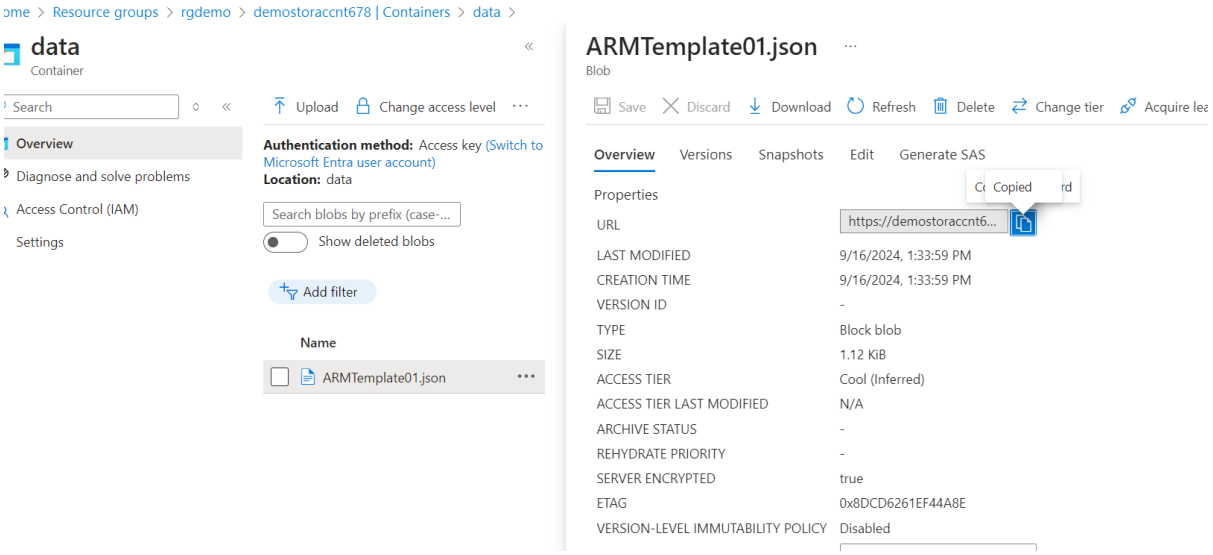
ARM Template Script to create web app and app service plan instance in Azure account.

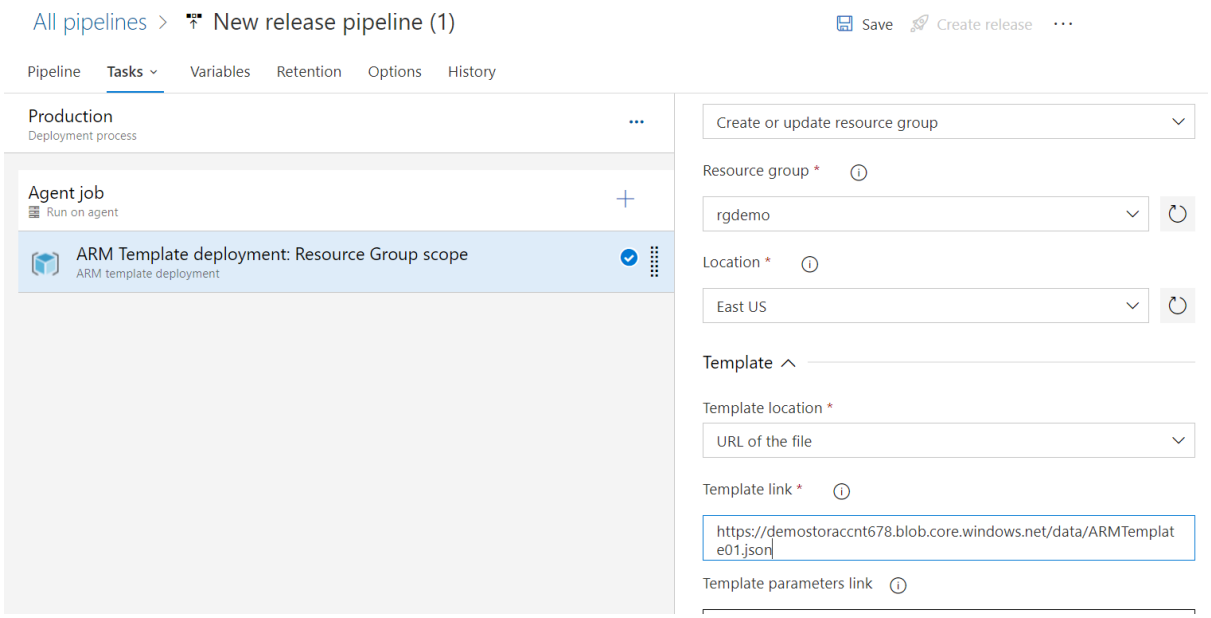
```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {},
  "functions": [],
  "variables": {},
  "resources": [
    {
      "type": "Microsoft.Web/serverfarms",
      "apiVersion": "2022-03-01",
      "name": "demowebapp09876",
      "location": "[resourceGroup().location]",
      "sku": {
        "name":"F1",
        "capacity":1
      },
      "properties":{
        "name":"demowebapp09876"
      }
    },
    {
      "type": "Microsoft.Web/sites",
      "apiVersion": "2022-03-01",
      "name": "websitedemo567",
      "location": "[resourceGroup().location]",
      "properties":{
        "name":"websitedemo567",

"serverFarmId":"[resourceId('Microsoft.Web/serverfarms','demowebapp09876')]"
      },
      "dependsOn":[
        "[resourceId('Microsoft.Web/serverfarms','demowebapp09876')]"
      ]
    }
  ],
  "outputs": {}
}
```

Navigate to Azure DevOps Portal and complete the configuration as per screenshot.

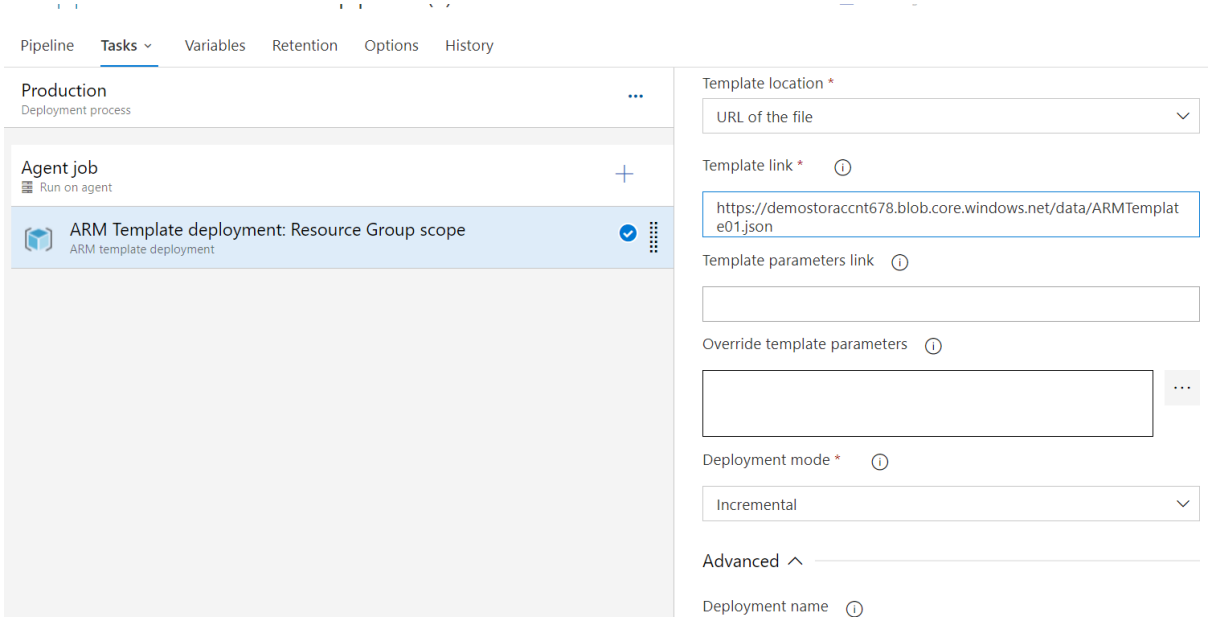


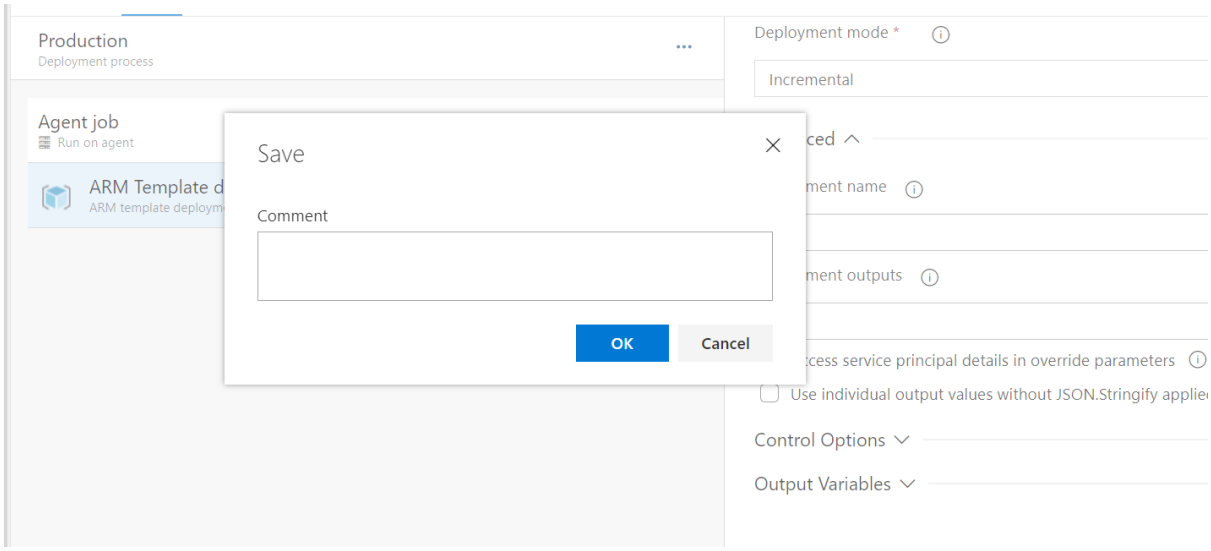
Paste the Url of Blob in the template link



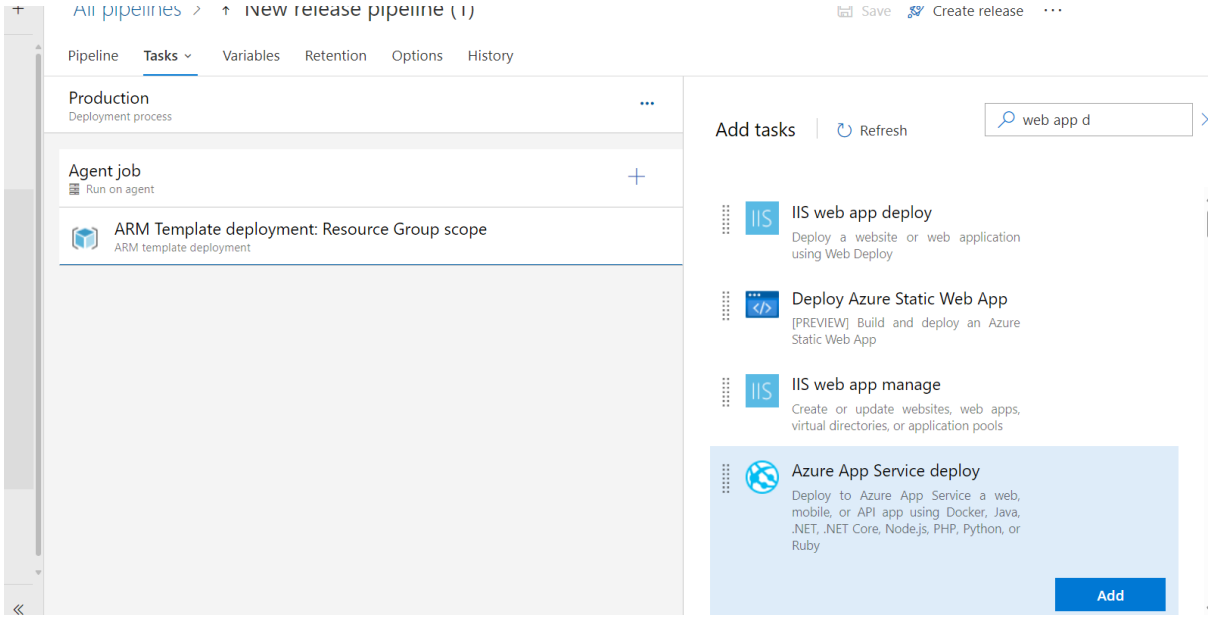


Rest all settings keep as default.

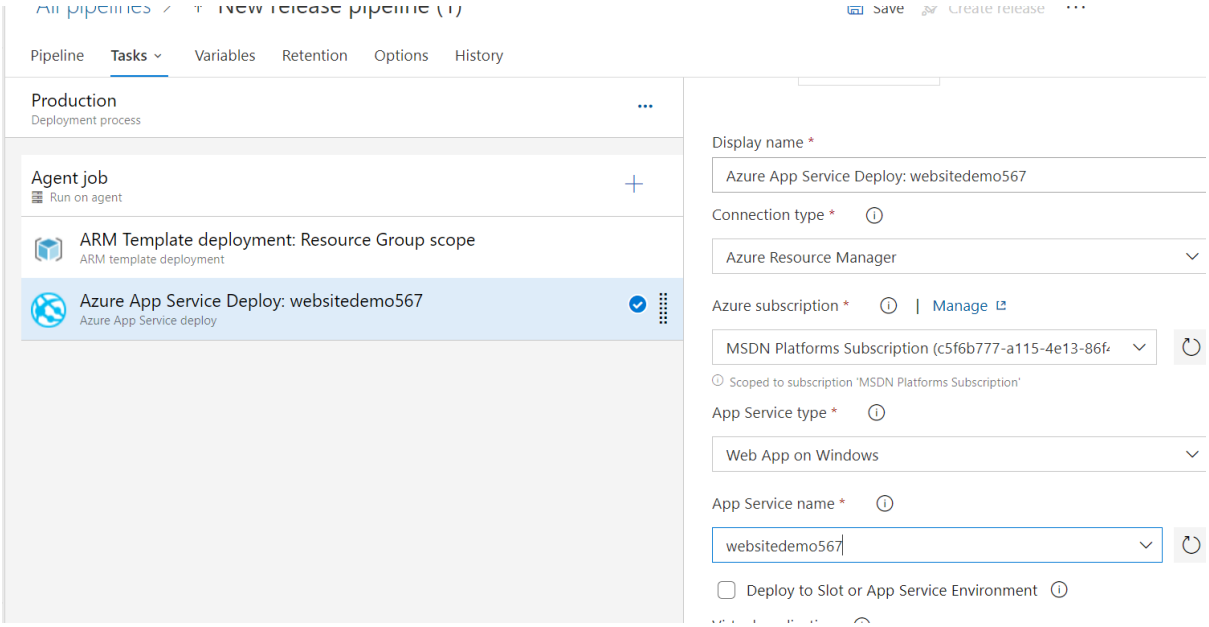




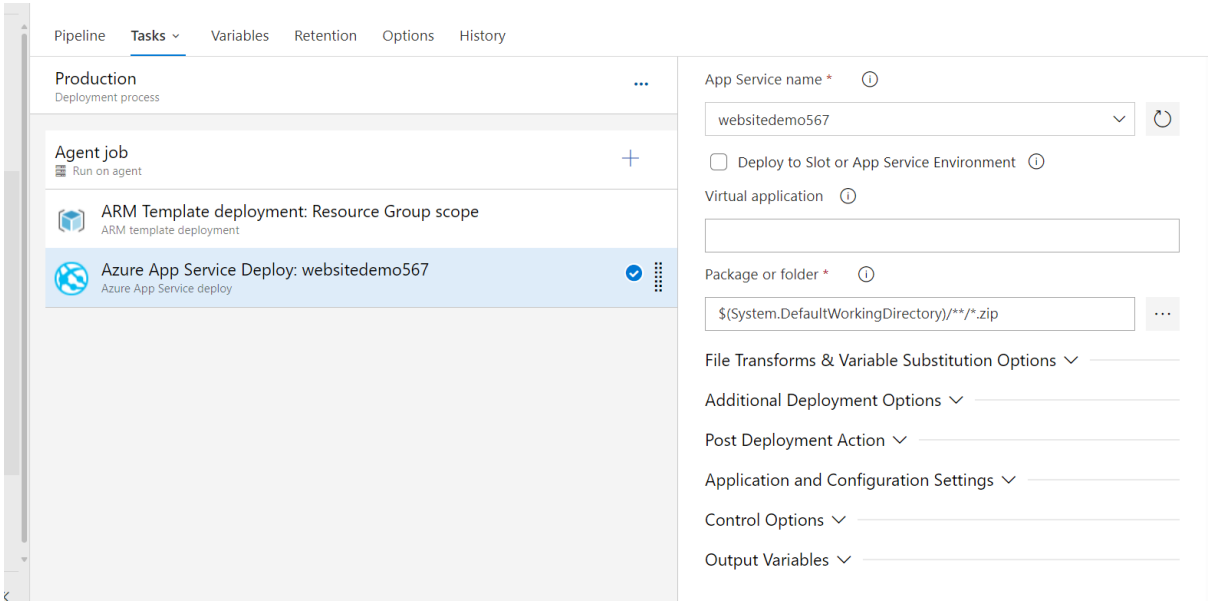
Add Task App Service Deploy and configure the settings in order to deploy our application to web app instance that will be created by ARM Template Deployment mentioned in Task1.

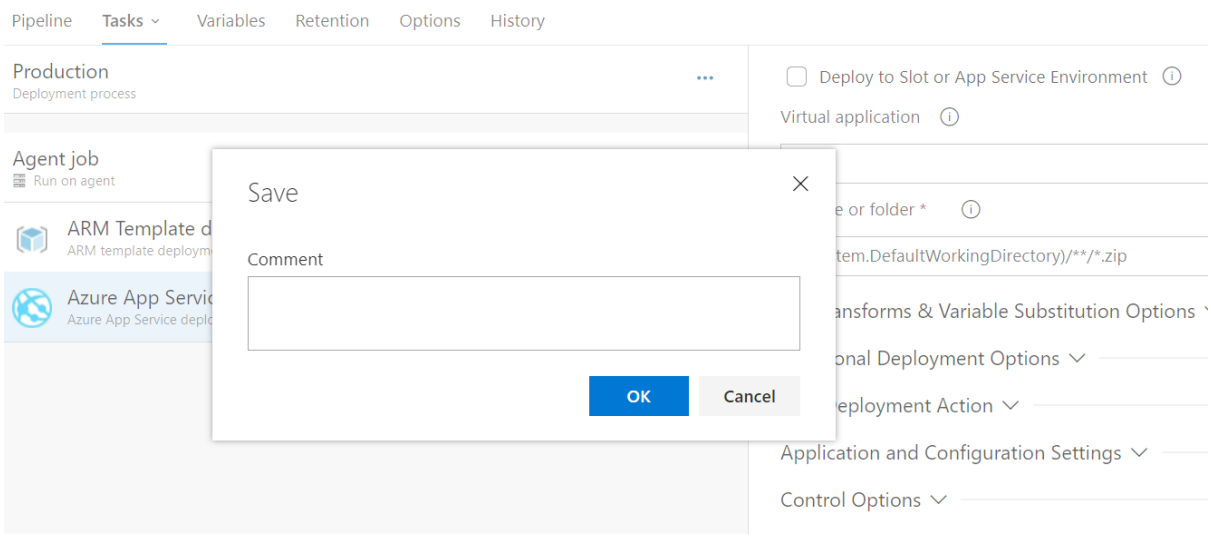


Paste the app service name from the script which you have uploaded to Azure storage account as blob.

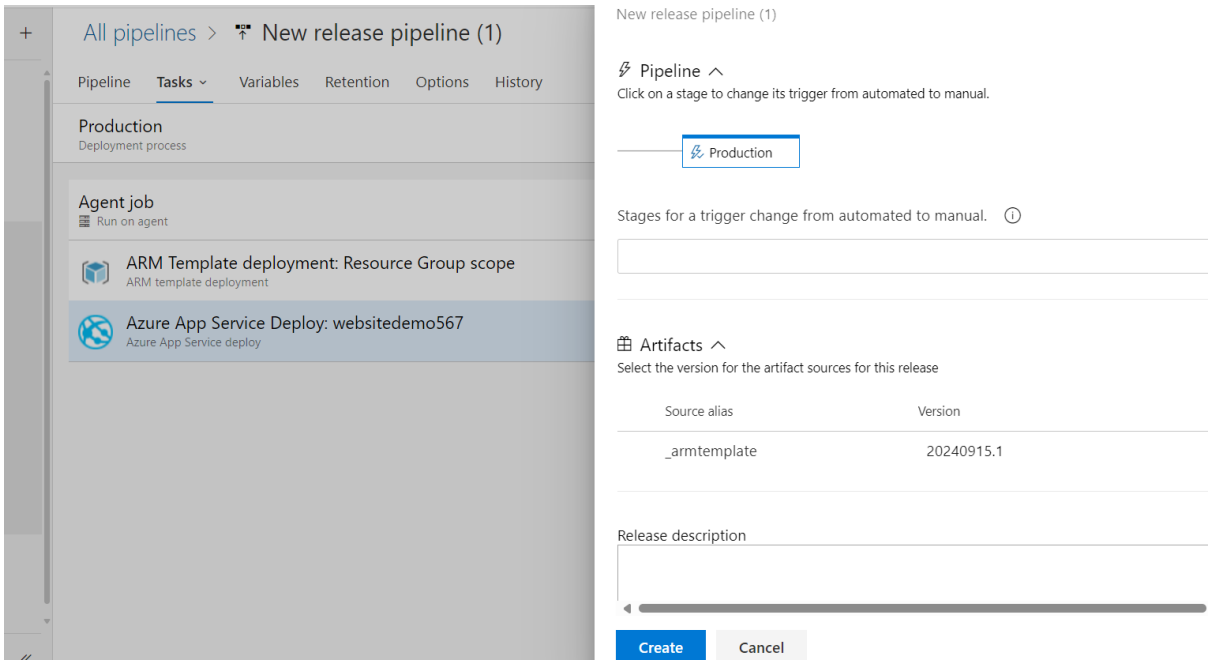


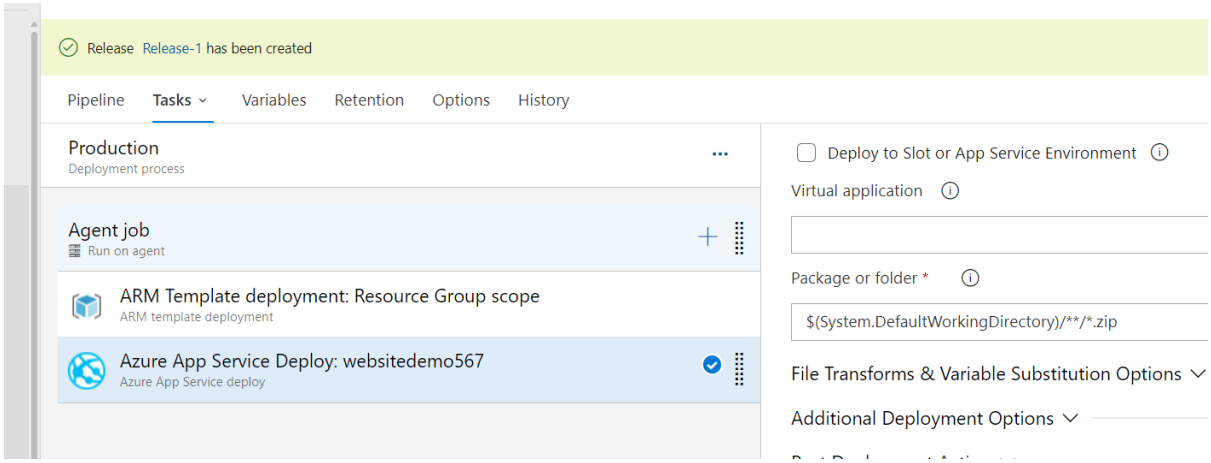
Rest other settings as default. Save it.



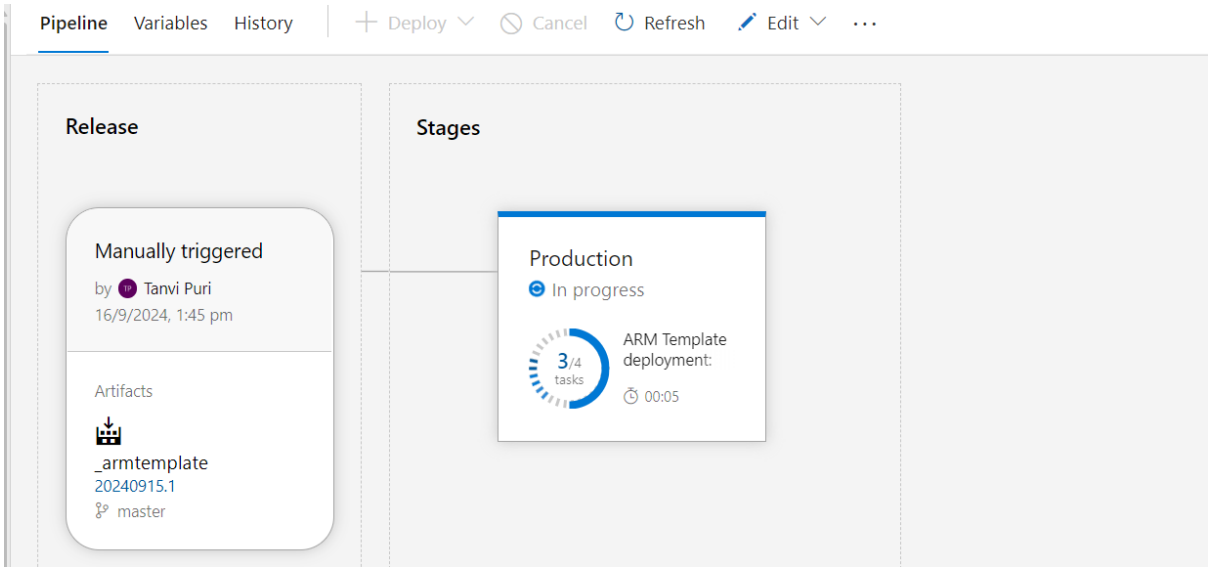


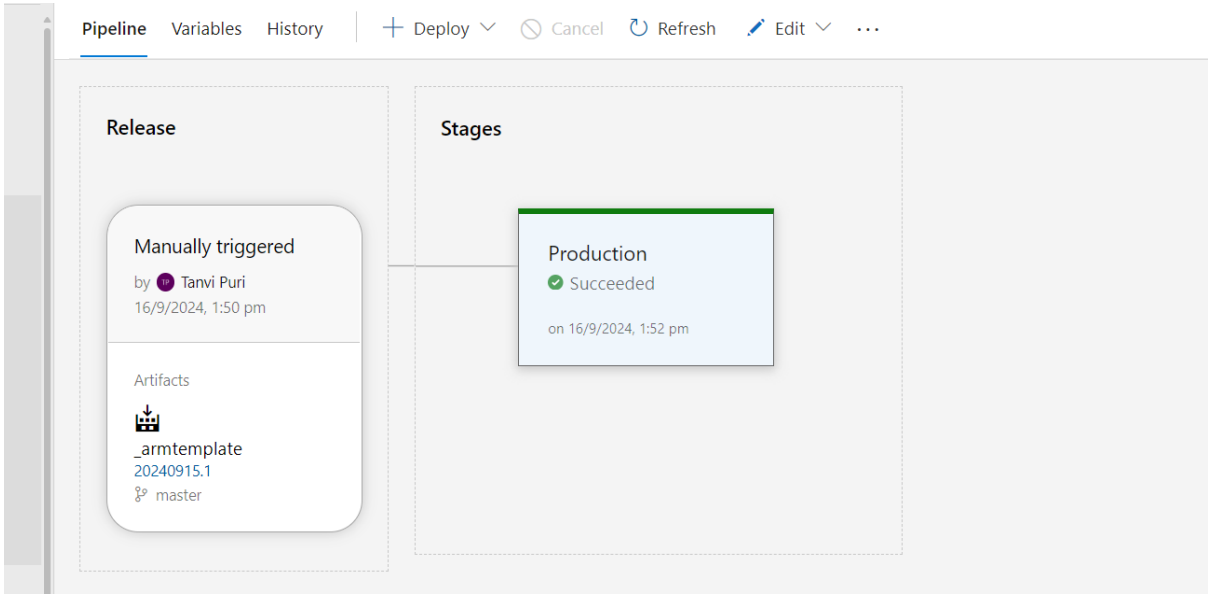
Create a Release Pipeline.





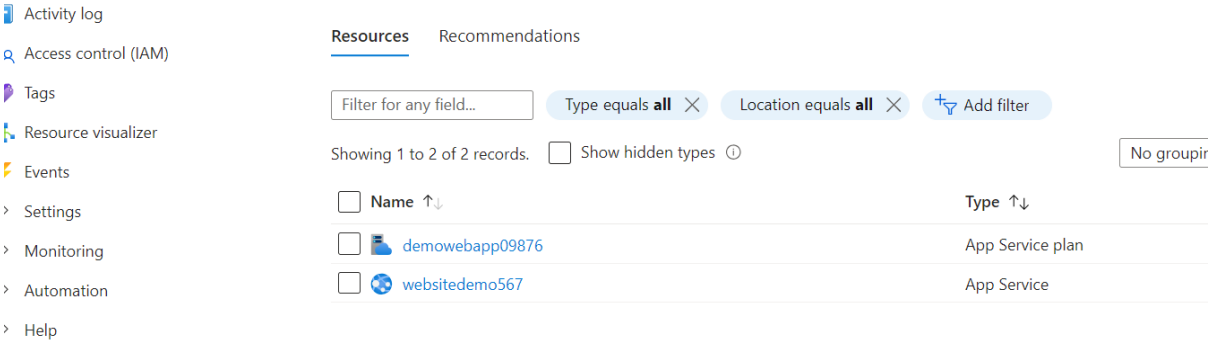
Click on Release and check the status.



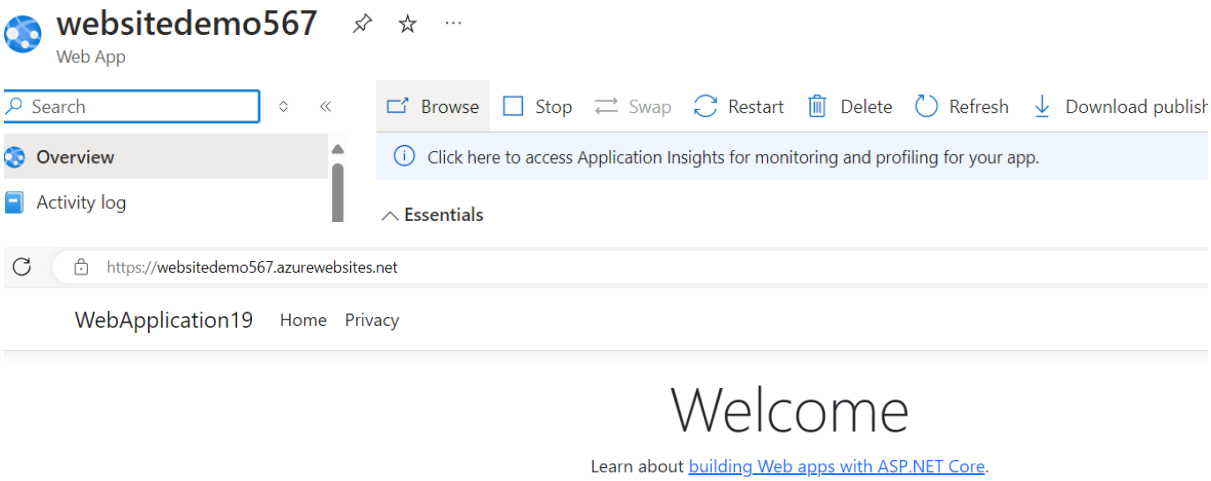


In case you get an error in pipeline execution try with some other region and SKU [S1 instead of F1].

Verify on Azure Portal



Hit on browse and see the deployment.


websitedemo567
Web App
Search
Browse
Stop
Swap
Restart
Delete
Refresh
Download publish
Overview
Activity log
Click here to access Application Insights for monitoring and profiling for your app.
Essentials
https://websitedemo567.azurewebsites.net
WebApplication19
Home
Privacy
Welcome
Learn about building Web apps with ASP.NET Core.