Project

**Agile Planning Using Azure Boards and Continuous Delivery with Azure Pipelines**

–

Table of Contents

[**Microsoft Azure + Storage Account + Azure Function + MVV** 1](file:///C:\onedrivetecpledgeconsulting\OneDrive%20-%20tpcsacademy\MShandbook\AZ-204T00A-Trainer-Files-Developing-solutions-for-Microsoft-Azure\Azure%20Function%20Project%20With%20Storage%20Queue.docx#_Toc54353827)

[**Microsoft Azure + Storage Account + Azure Function + MVV** 1](file:///C:\onedrivetecpledgeconsulting\OneDrive%20-%20tpcsacademy\MShandbook\AZ-204T00A-Trainer-Files-Developing-solutions-for-Microsoft-Azure\Azure%20Function%20Project%20With%20Storage%20Queue.docx#_Toc54353828)

[**1.** **The Project Summary** 3](#_Toc54353829)

[**2.** **The Proposed Solutions (Logical Architecture)** 3](#_Toc54353830)

[**3.** **The Proposed Test-bed Scenario and High Level Steps:** 3](#_Toc54353832)

[**4.** **The Technical Execution Steps** 4](#_Toc54353833)

[Step1: Azure Portal Sign-in and Create Required Resources 4](#_Toc54353834)

[**Task 1: Sign in to the Azure Portal** 4](#_Toc54353835)

[**Task 2: Create the Resource Group** 4](#_Toc54353836)

[**Task 3: Create the Storage Account** 5](#_Toc54353837)

[**Task 4: Create a queue for input messages** 6](#_Toc54353838)

[Step 2: Create Azure Function using visual studio 7](#_Toc54353839)

[**Task 1: Create Azure Function App** 7](#_Toc54353840)

[**Task 2: Create Azure Function with queue trigger** 9](#_Toc54353841)

[**Task 3: Publish the function to Azure** 11](#_Toc54353842)

[**Task 4: Verify the functionality** 16](#_Toc54353843)

# **The Project Summary**

**Project Title** :

Agile Planning Using Azure Boards and Continuous Delivery with Azure Pipelines

**DESCRIPTION**

You work as a senior DevOps engineer in DevTech Inc. Your organization wants to automate the deployment of applications. The development and operations team wants to use Azure DevOps to configure and automate deployments of their builds to target stages easily. They decided to work on the project with DevTechOps to verify the functionality.

The DevOps team at DevTech decides to define the application infrastructure and use the Azure Pipelines to deploy the application. They want to plan, manage, and track the work using Azure boards.

**Background of the problem statement:**

DevTech Inc. is developing an end-to-end pipeline for a retail e-commerce web application that will sell customized products to its customers in the US. They decide to deploy the application on Azure app services. They want any new changes pushed to Azure Repos to trigger the release automatically. To track the development task, they want to use Azure Boards. The team also wants to leverage Azure Repos for storing code and creating Repositories. The DevOps team expects to create a continuous release for the QA and production stages.

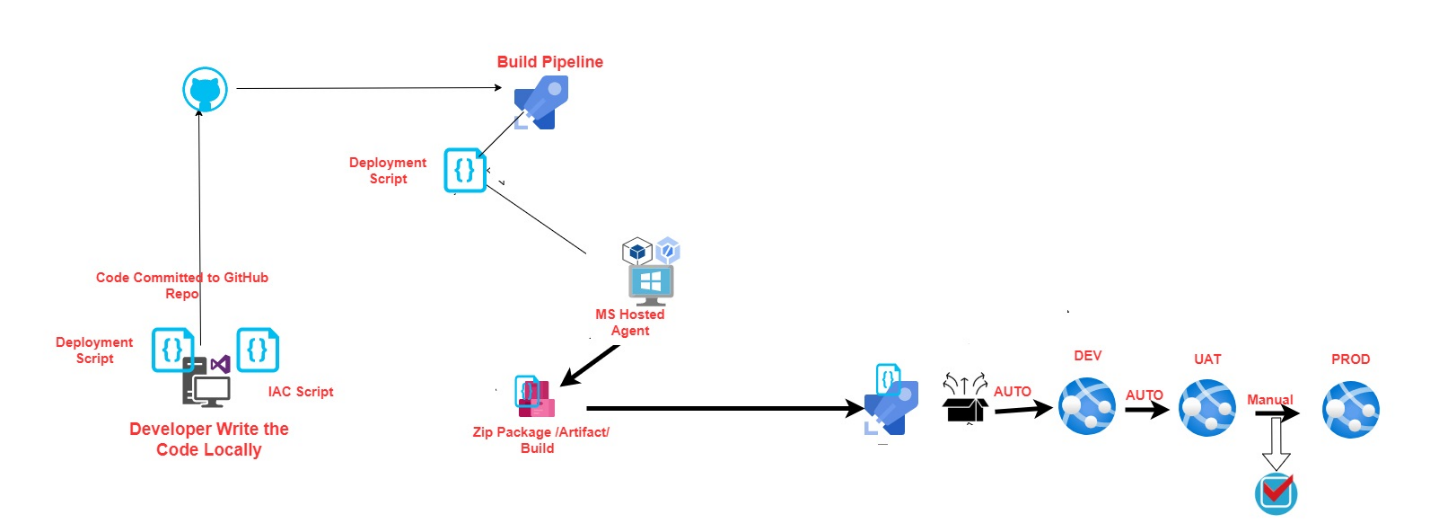
As a security measure, you need to create a specific group for the DevOps team and assign roles to them. This ensures the team can deploy the resources mentioned in the requirements below and adhere to the principle of least privilege.

As part of compliance, the DevOps team wants all resources to be created in the US East and the US West regions. They want to use Azure Policy to allow specific regions.

The solution should meet the following requirements:

* + - Define Azure Policy to allow the creation of resources in a specific region
    - Create a group for the DevOps team and define access using RBAC
    - Define the application infrastructure using app services
    - Create Azure work items in Boards to track work
    - Create a repository in Azure Repos
    - Configure the Azure app service
    - Create a continuous release to the QA stage
    - Create a release to the production stage
    - Test the deployment of the web application

# **The Proposed Solutions (Logical Architecture)**

****

# **High Level Steps:**

**The High Level Execution Steps**

To create the full CI/CD deployment using Azure DevOps Services including Azure Boards and Azure pipeline we are performing following steps

1. Log in to the Azure Portal with Admin Privilege
2. Create Following resources
   1. Resource Group
      1. Create a Resource group for the DevOps team and define access using RBAC
      2. Create App services and two additional deployment slots named **Dev** and **UAT**.
      3. Login to DevOps Portal
      4. Create Azure work items in Boards to track work
      5. Create a repository in Azure Repos
      6. Configure the Azure app service
      7. Create a continuous release to the QA stage
      8. Create a release to the production stage
      9. Test the deployment of the web application

# **The Technical Execution Steps**

## Step1: Azure Portal Sign-in and Create Required Resources

### **Task 1: Sign in to the Azure Portal**

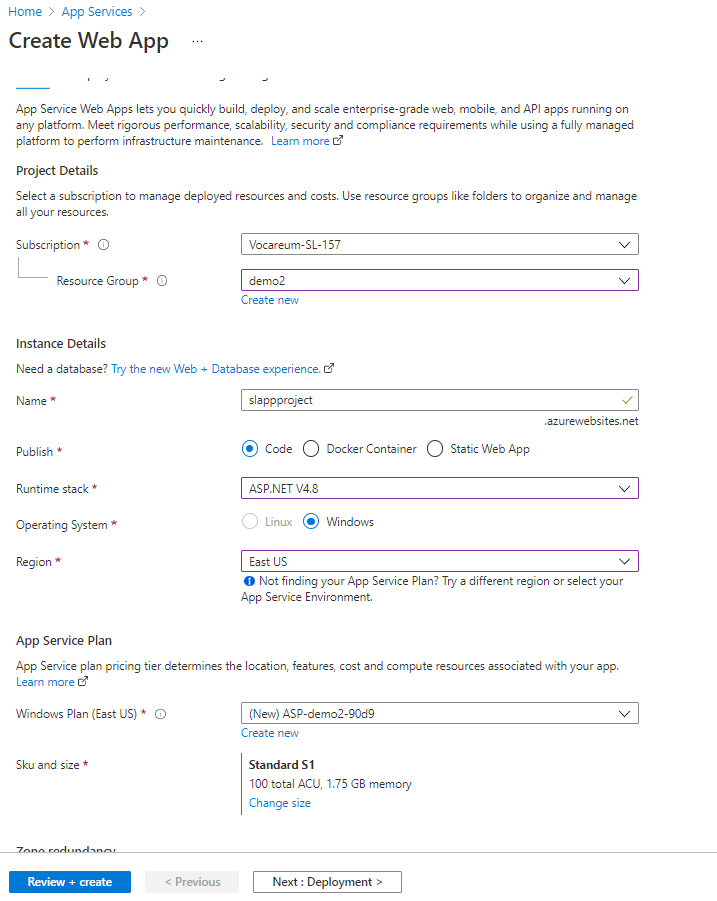
1. On the Start screen, click the **Internet Explorer** tile.
2. Go to *(*[*https://portal.azure.com*](https://portal.azure.com)*)*.
3. Enter the email address of your Microsoft account. Click **Next**.
4. Enter the password for your Microsoft account.
5. Click **Sign In**.

### **Task 2: Create the Resource Group**

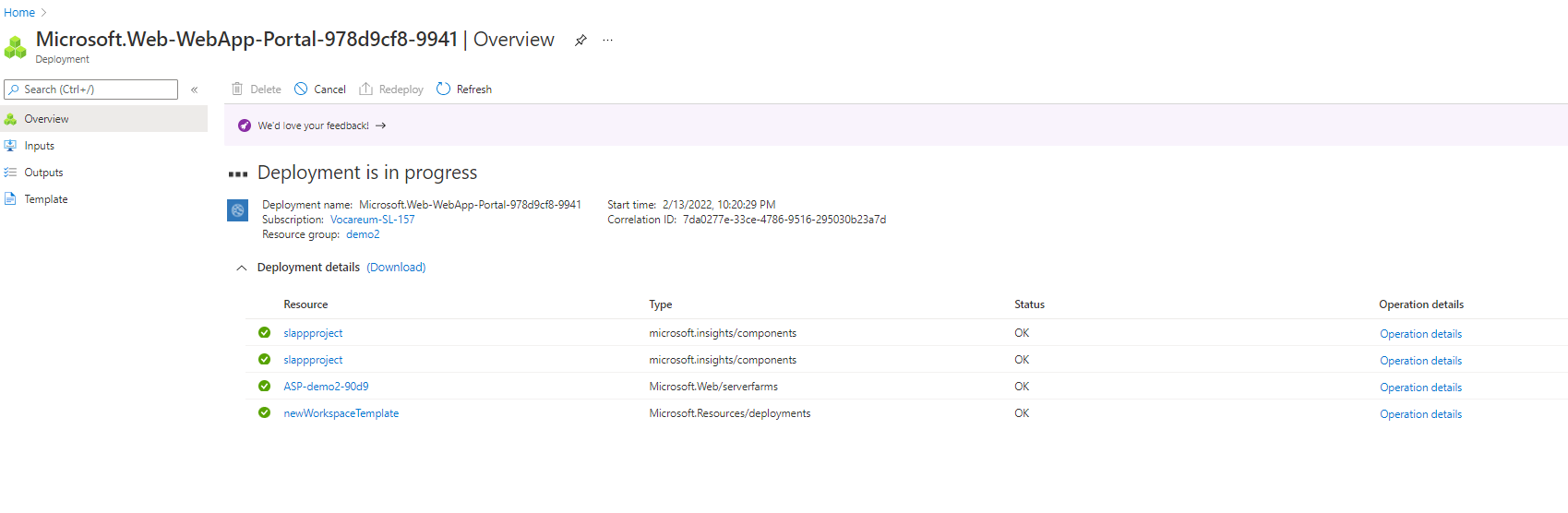
1. In the navigation pane on the left side of the Azure Portal, click **All services**.
2. In the **All services** blade that displays, click **Resource groups**.
3. In the **Resource groups** blade that displays, view your list of resource groups.
4. At the top of the **Resource groups** blade, click the **Add** button.
5. In the **Resource group** blade, perform the following steps:
   1. In the **Resource group name** dialog box, provide the value **InGenRG**.
   2. In the **Resource group location** list, select **East US**.
6. In the **Resource group** blade, click **Create**.

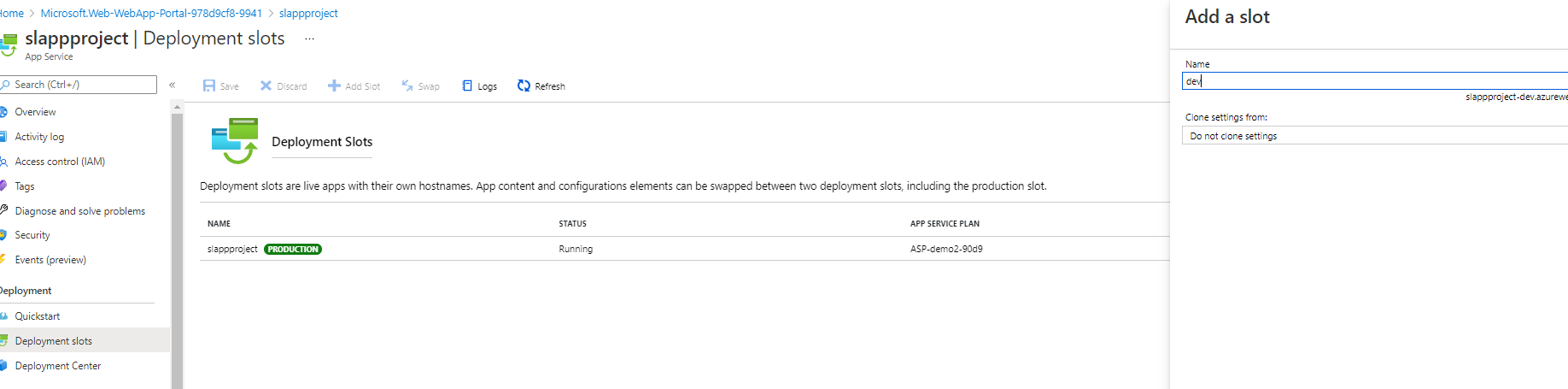
### **Task 3: Create the App Service**

1. In the navigation pane on the left side of the Azure Portal, click **All services**.
2. In the **All services** blade that displays, click **App Service**
3. Create an service



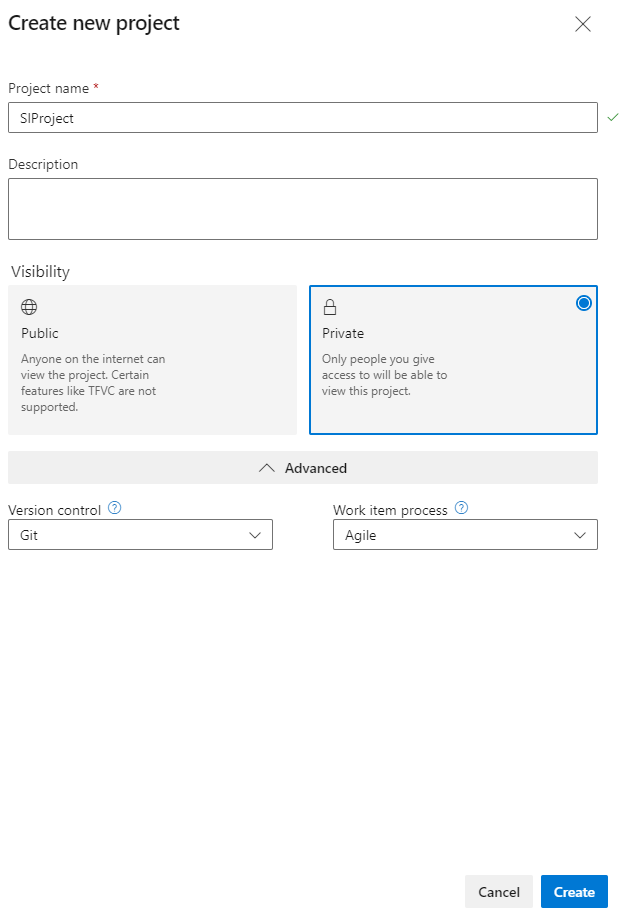
1. Deployment started

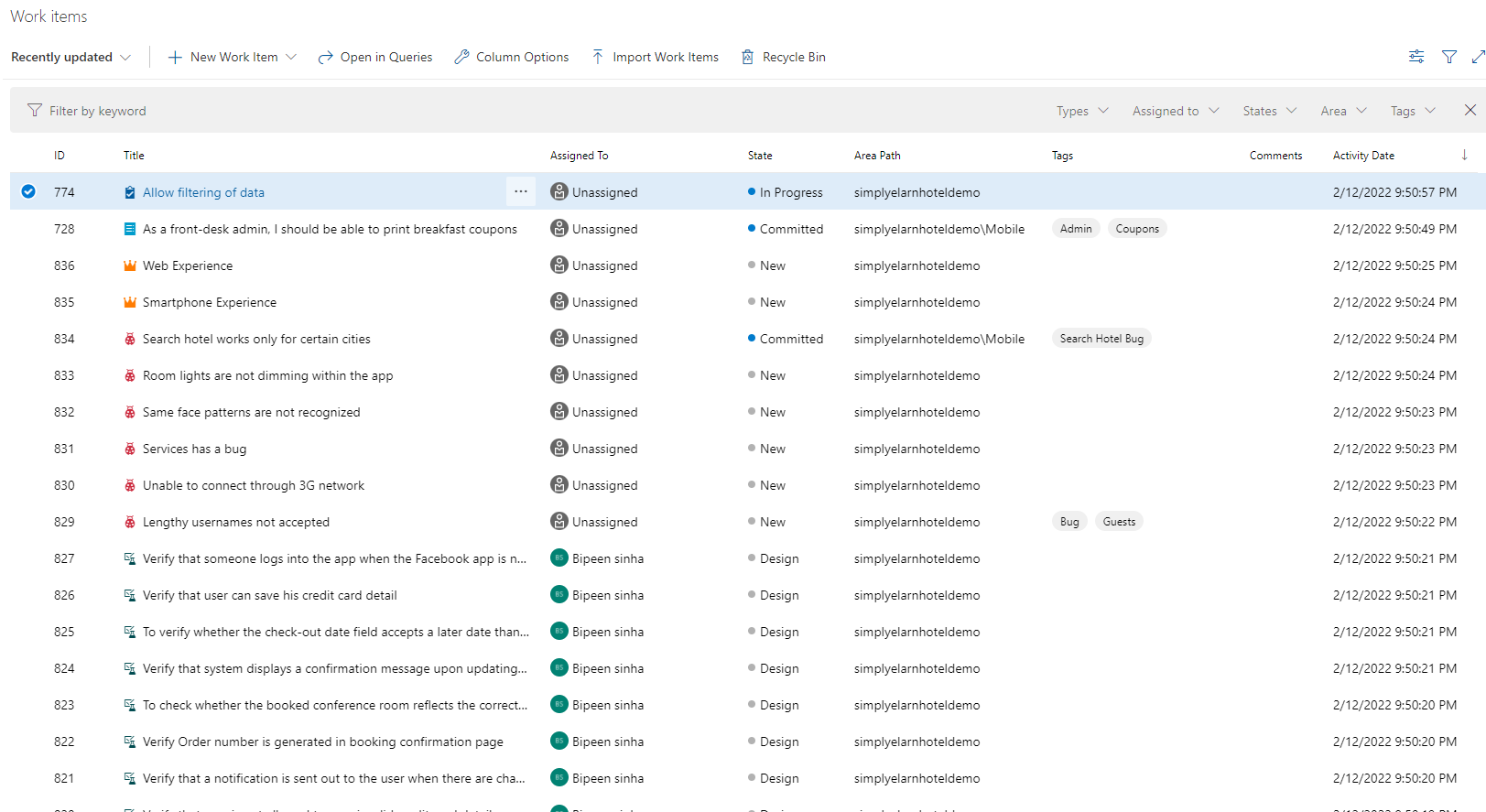


1. Once Deployment Complete go to resource
2. Click Deployment slots and create Dev and UAT Slots

### **Task 4: Create a Work Item in Azure DevOps**

1. Login to the azure devops
2. Create New project

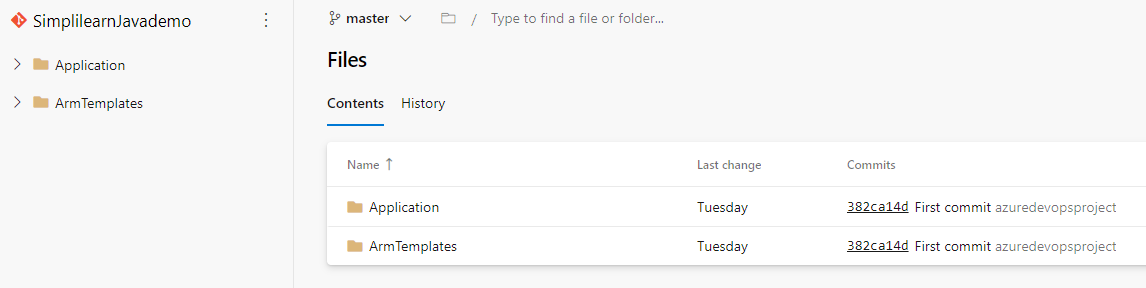


1. Go to Board and Create Work item as shown in picture

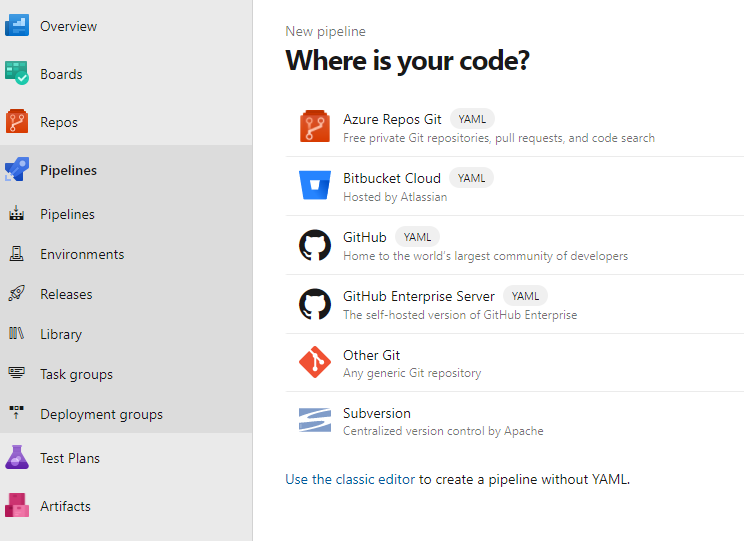
## Step 2: Create Azure Build and Release Pipeline

### **Task 1: Create Azure Build Pipeline**

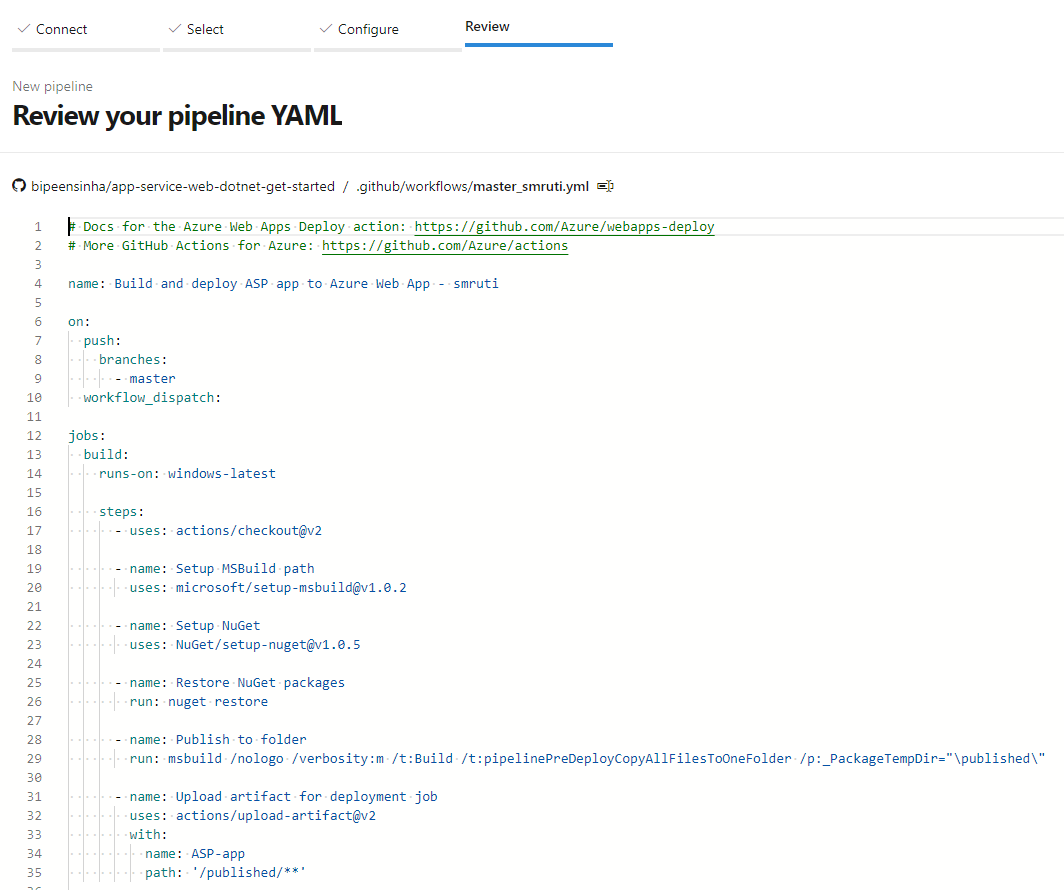
1. Clone the Azure Repo in your local Machine’
2. Upload the sample code in azure Repo



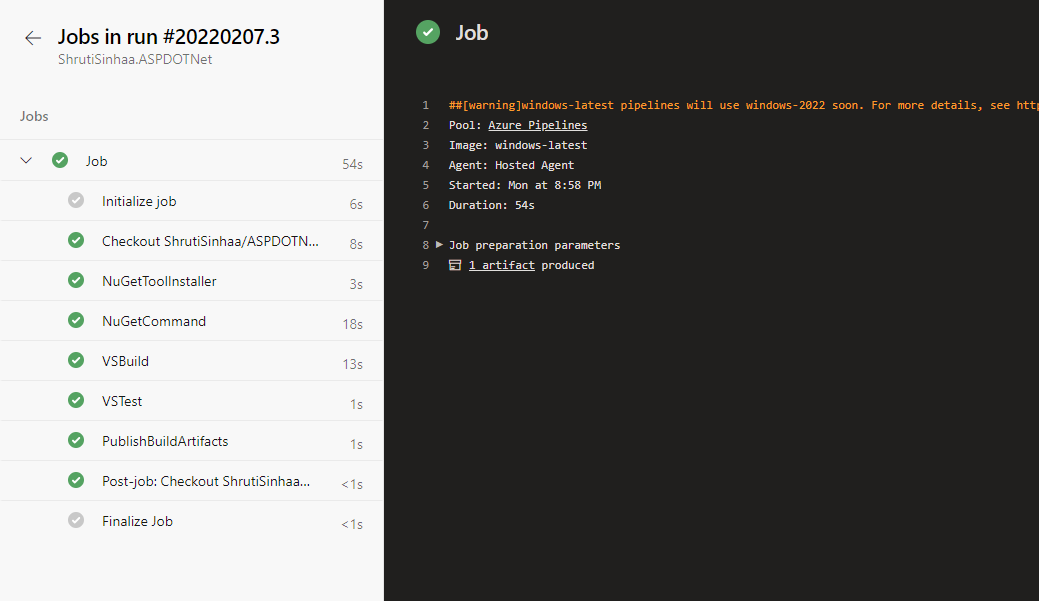
1. Go to Pipeline and Create New Build Pipeline
2. Select Azure Repo with YAML



1. You should see your YAML as below

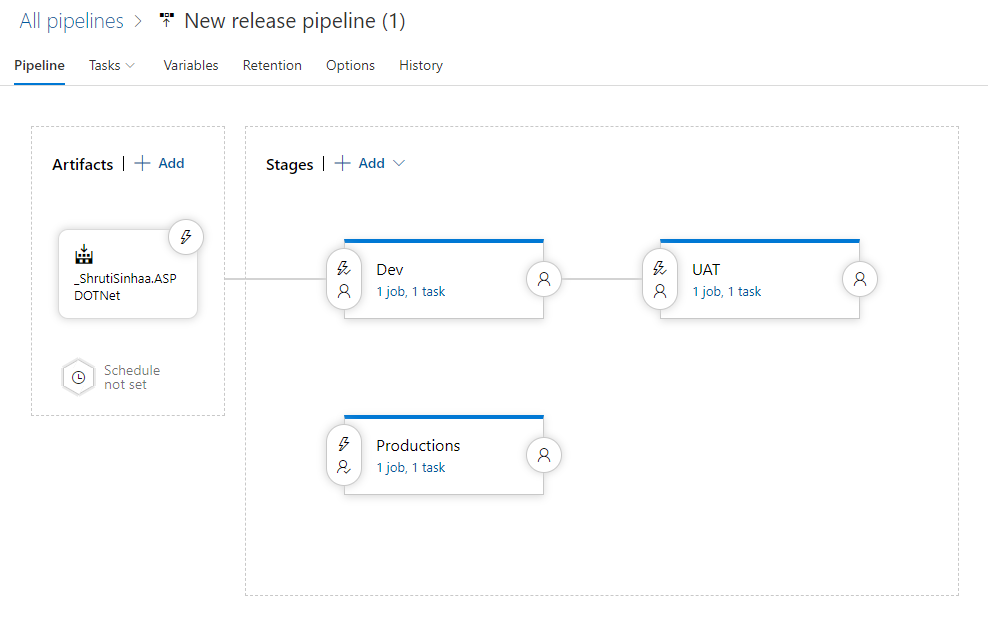


1. Now Click Run and Go to the Job.
2. You should see the Pipeline is now building the code

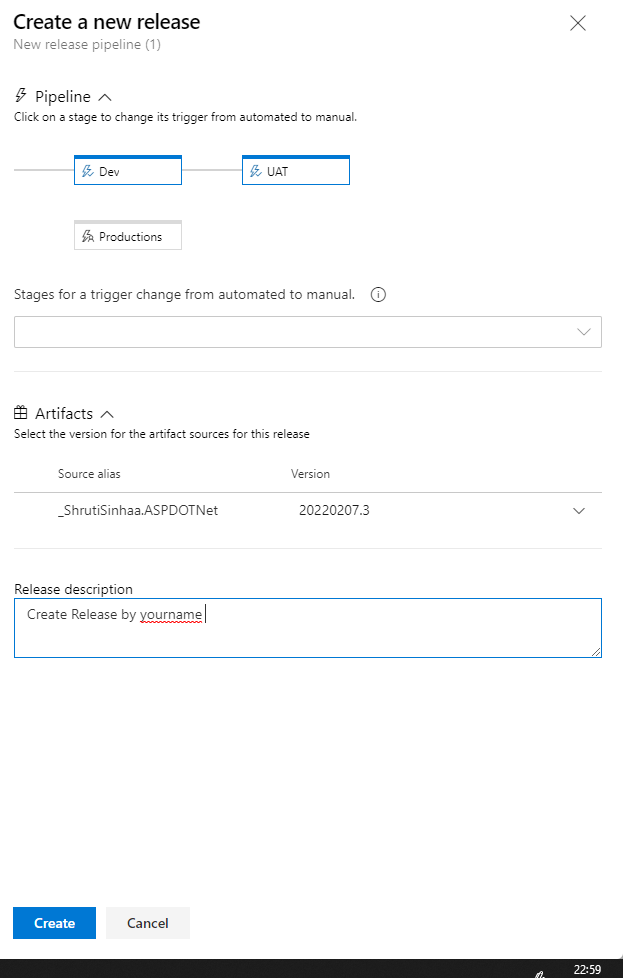


### **Task 3: Create Release Pipeline**

1. Create Release Pipeline as shown below Picture



1. Create Release



1. Click Release and Notice the deployment automatically started for Dev and UAT but in Prod the deployment begin only after approval

