

**Module 4 Lab: VSTS CI/CD with App Services**

Template Version: 2.0

**Estimated Time**

60 minutes

**Objectives**

At the end of this lab, you will be able to:

* Setup a CI/CD pipeline on your own for a simple web app.
* Make changes and see them appear in Azure after you push changes to VSTS.

**Logon Information**

Please use the Azure Pass provided to you for this lab. Your VM credentials are listed for later use:

* VM Username: super
* VM Password: P@ssw0rd123!

Table of Contents

[Module 4 Lab: VSTS CI/CD 3](#_Toc523182040)

[Exercise 1: VSTS CI (Build Definition) 3](#_Toc523182041)

[Exercise 2: VSTS CD (Release definition) 7](#_Toc523182042)

[Exercise 3: End to End Testing – Putting it All Together 13](#_Toc523182043)

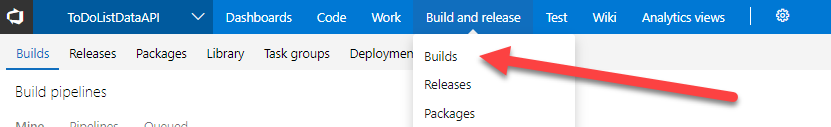
Module 4 Lab: VSTS CI/CD

Exercise 1: VSTS CI (Build Definition)

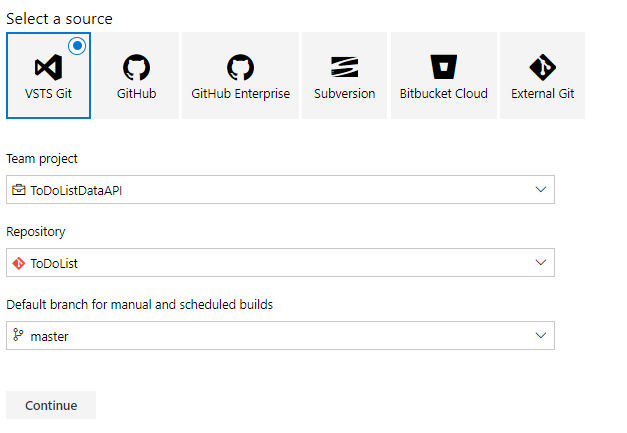
This exercise shows how to use your Azure account.

Tasks

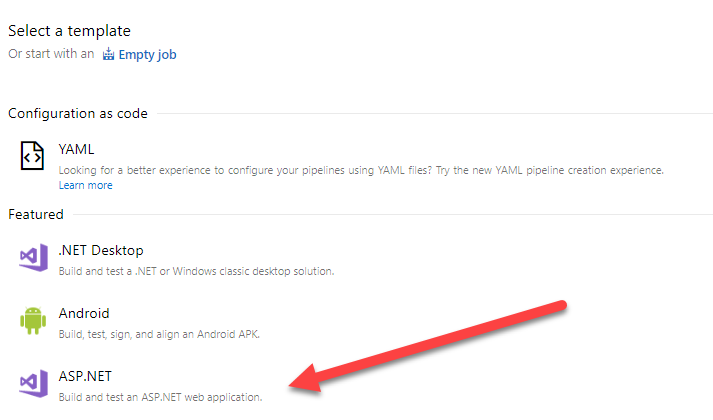
1. Login to your VSTS account
2. Browse to initials-occ.visualstudio.com and login with your personal Outlook account associated with your personal Azure account.
3. Click on the Project that you created.
4. Click on the Build and Release tab, then Select Builds.



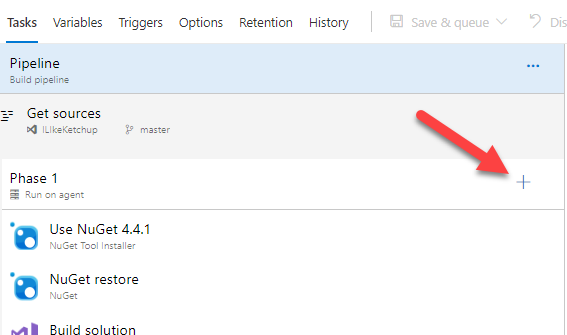
1. **Create a new build definition**
2. Click “+ New” to create a new build definition.
3. On the first page, select VSTS Git, your team project, repository, and the master branch. Don’t worry! You might have different names for these than I chose, but you should only have one to choose from 😊.



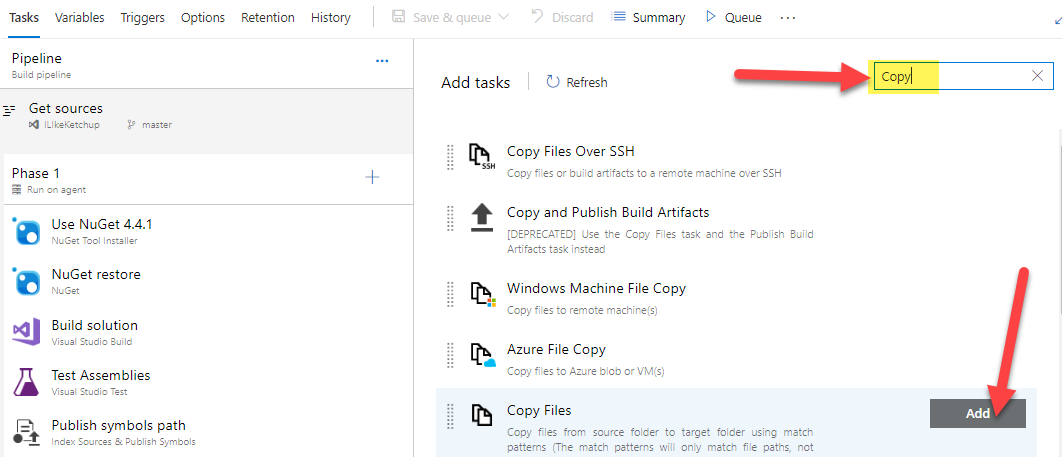
1. Click Continue.
2. When asked to select a template, choose “ASP.NET”.



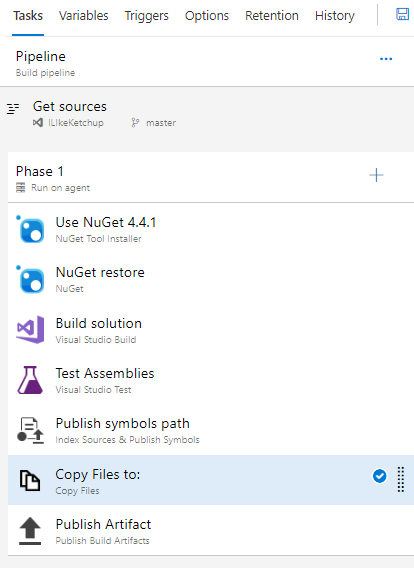
1. Add a task by clicking the “+” sign next to *Phase 1*.



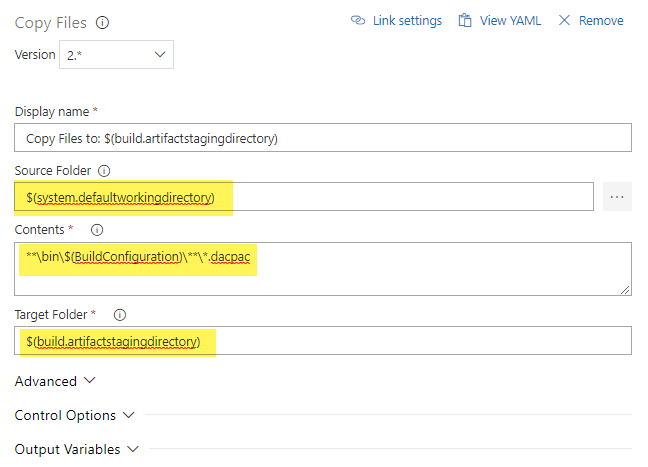
1. Search for “Copy”. Mouseover the **Copy Files** task and click the **Add** button.



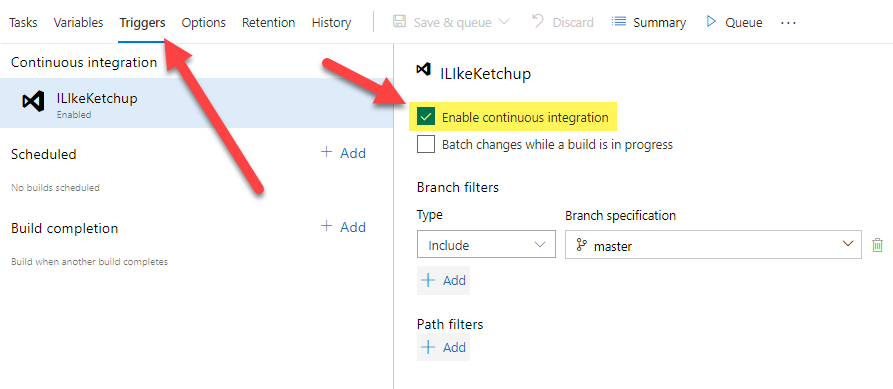
1. Click on the **Copy** task and drag and drop it to be before the **Publish Artifact** task.



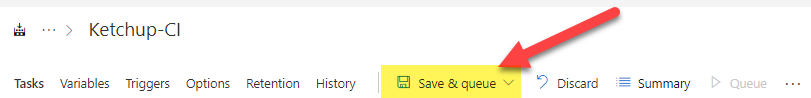
1. This new task will copy the .dacpac file so we can publish our database schema to Azure SQL. Click on the **Copy Files** task and fill in the following:
   1. Source Folder: $(system.defaultworkingdirectory)
   2. Contents: \*\*\bin\$(BuildConfiguration)\\*\*\\*.dacpac
   3. Target Folder: $(build.artifactstagingdirectory)



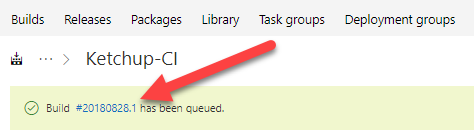
1. Click on the Triggers tab. Check the box for **Enable continuous integration.**



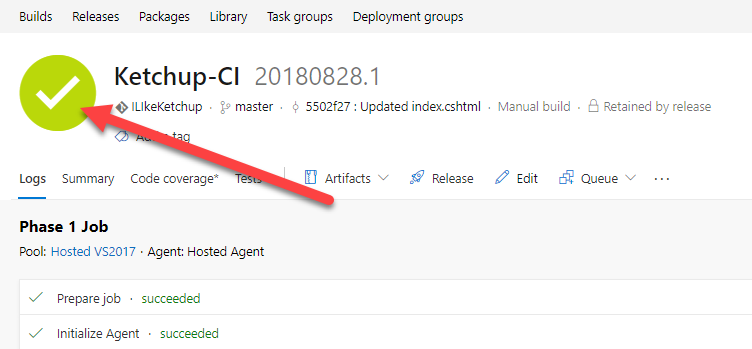
1. Click Save & Queue.



1. On the modal pop-up, click Queue and leave everything as default.
2. Click on the blue text that shows up near the top of the screen with the build #.



1. Wait for a success (green check icon shown below), it should look like below. If you do not get a success, double check your work and debug based on the error or get help from your instructor.



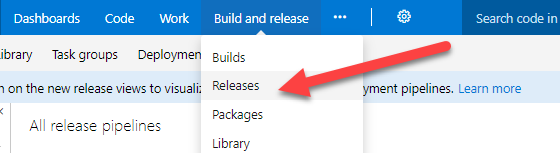
Exercise 1 has been completed

Exercise 2: VSTS CD (Release definition)

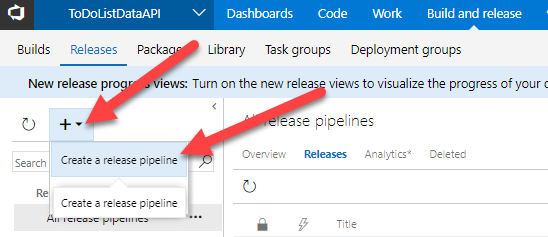
This exercise shows the point in time restore of the database as a different database.

Tasks

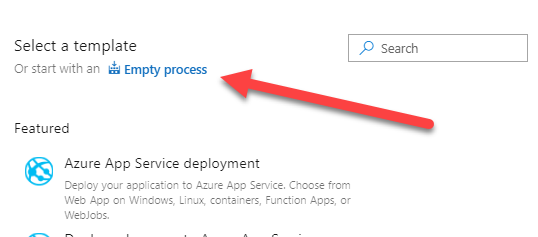
1. Create a new release definition
2. Click on **Build and release**, then choose **Release**.



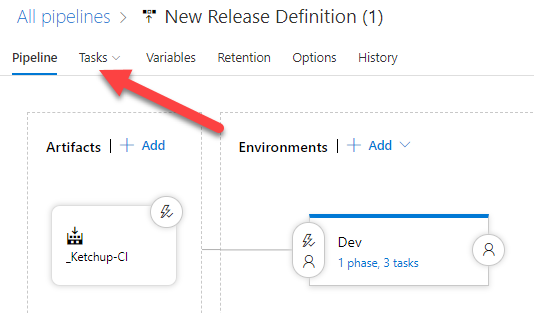
1. Click “+”, then **Create a release pipeline**.



1. Choose the **Empty process**.



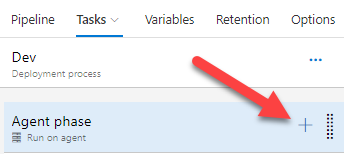
1. Click the **Tasks** tab.



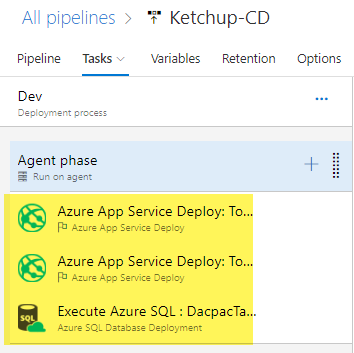
1. Change the name of your Release definition by clicking on the text on the top.



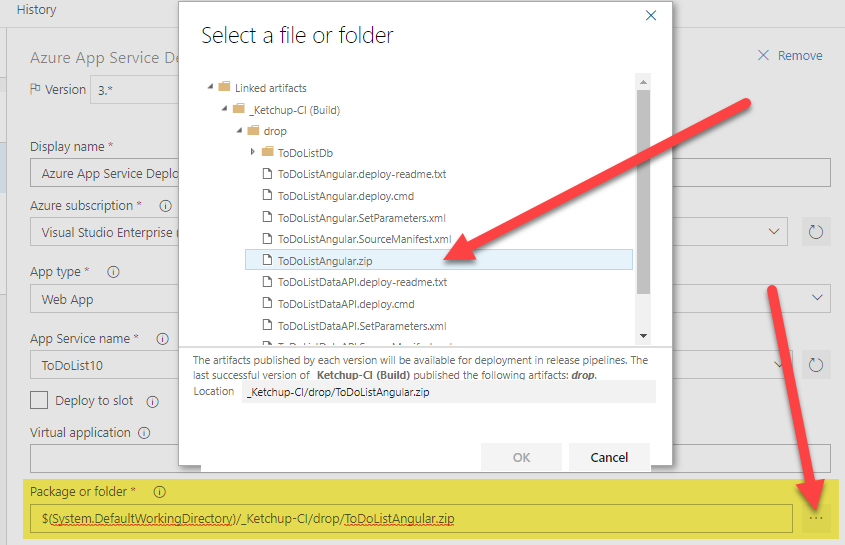
1. Click the “+” button to add a new task.



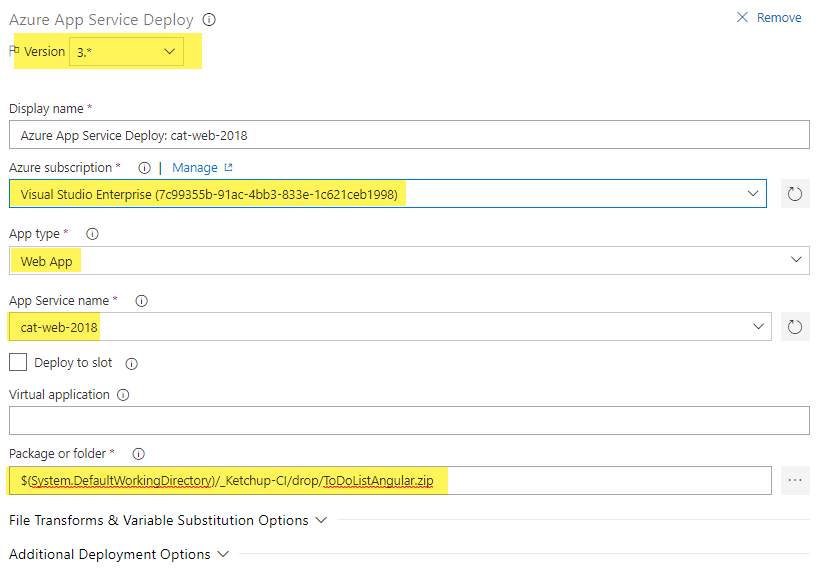
1. Search for and add the following 3 tasks:
   1. **2** x Azure App Service Deploy
   2. Execute Azure SQL
2. You should see the following Tasks now, just make sure the icons match and you have 3 that look similar:



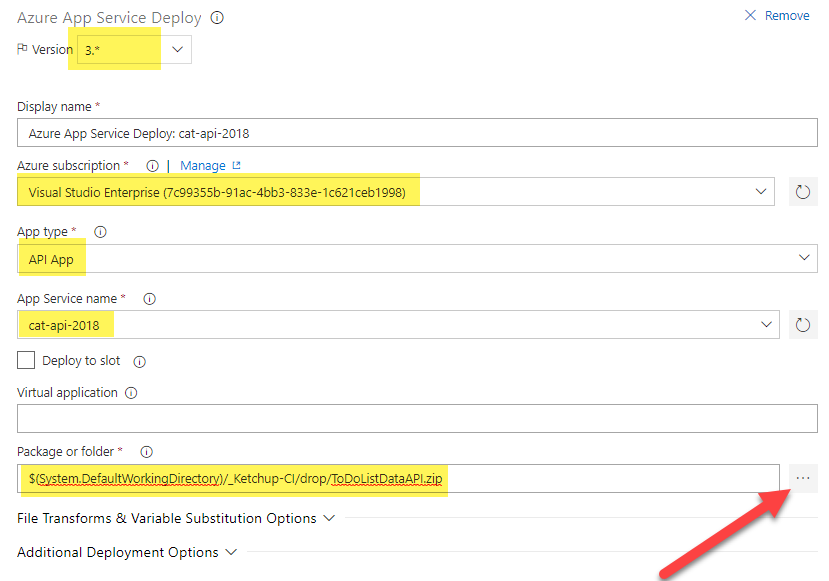
1. Click on the first Azure App Service task. Fill out the form as follows:
   1. Version 3.\*
   2. Azure subscription: Select your Azure Pass from the dropdown, click Authorize if needed.
   3. App type: Web App
   4. App Service Name: initials-web-2018
   5. Package or Folder: select ToDoListAngular.zip (screenshot below)



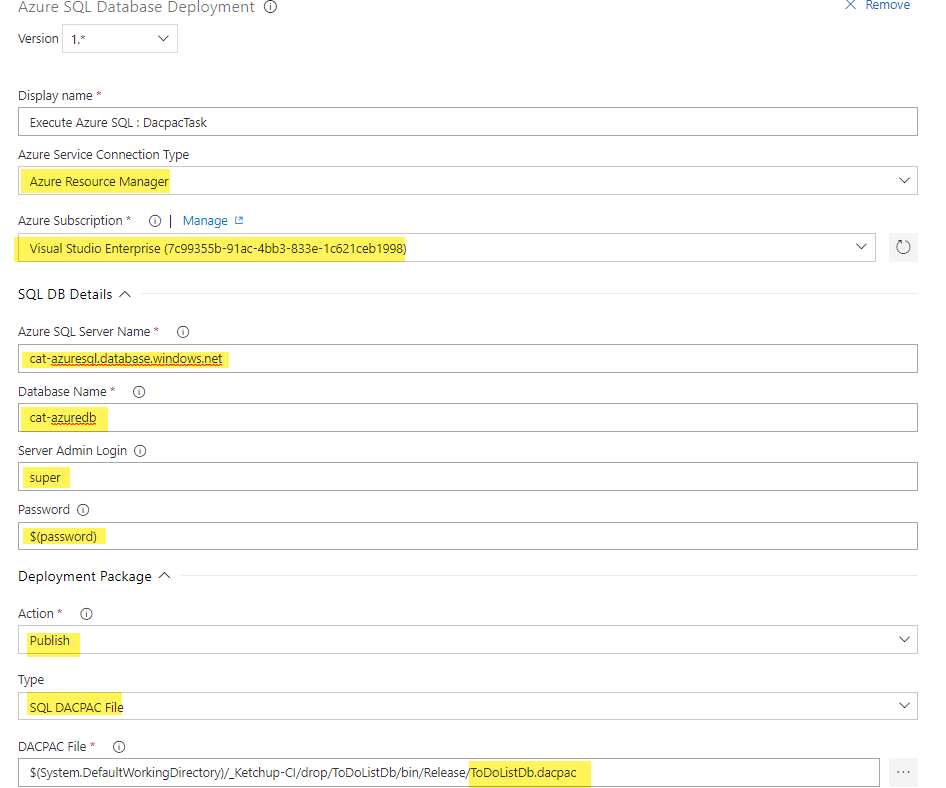
* 1. Verify that your task looks as follows, note your subscription will be Azure Pass though and screenshot shows Visual Studio Enterprise.



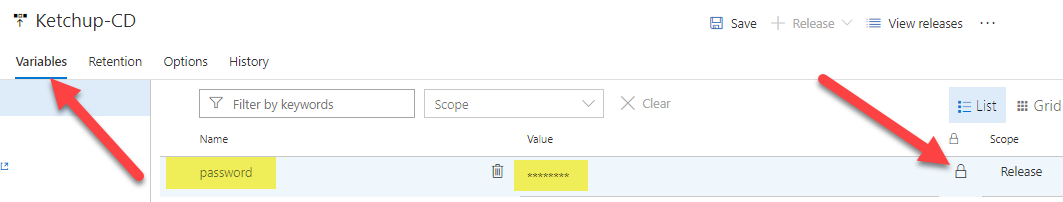
1. Click on the second Azure App Service task. Fill out the form as follows:
   1. Version 3.\*
   2. Azure subscription: Select your Azure Pass from the dropdown, click Authorize if needed.
   3. App type: API App
   4. App Service Name: initials-api-2018
   5. Package or Folder: select ToDoListDataAPI.zip
2. Verify that your task looks as follows, note your subscription will be Azure Pass though and screenshot shows Visual Studio Enterprise.



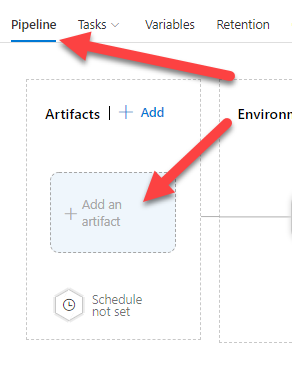
1. Click on the Execute Azure SQL task. Fill out the form as follows:
   1. Azure Service Connection Type: Azure Resource Manager
   2. Azure Subscription: Azure Pass
   3. Azure SQL Server Name: INITIALS-azuresql.database.windows.net
   4. Database Name: INITIALS-azuredb
   5. Server Admin Login: super
   6. Password: $(password)
   7. Action: Publish
   8. Type: SQL DACPAC File
   9. DACPAC File: locate the ToDoListDb.dacpac file
2. Verify your Execute Azure SQL task looks as follows:



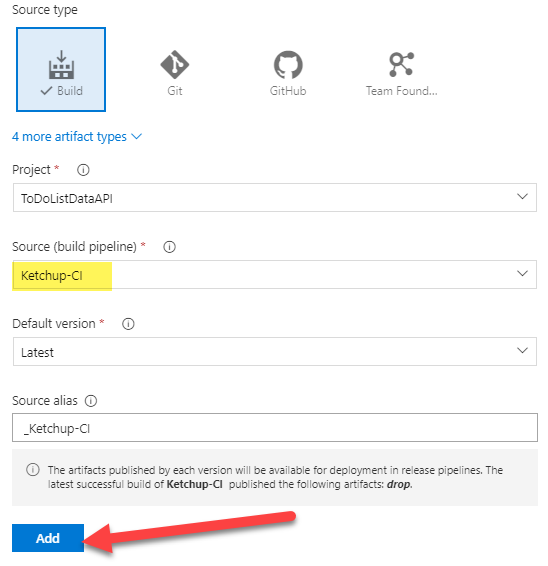
1. Click on the Variables Tab
   1. Add a variable with:
      1. Name: password
      2. Value: P@ssw0rd123!
      3. Click the lock icon



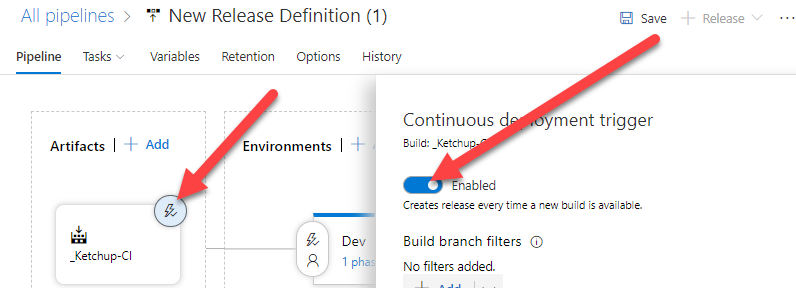
1. Click on the Pipeline view. Click Add an Artifact.



1. Add your CI build that you created in the last exercise. Click Add.



1. Click on the lightning bolt icon to add a Trigger. Click Enable.



1. Click Save.
2. Click Release > Create a Release. On the right pop-up, leave everything as default, and hit Create.

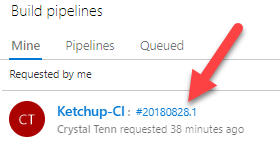
Exercise 2 has been completed

Exercise 3: End to End Testing – Putting it All Together

This exercise shows how to rename old and new databases

Tasks

1. Make Code Changes, Commit, and Push using Visual Studio
2. Try doing the next steps on your own without referencing another lab. If you need help with the following steps, please refer to **Module 3 lab** or let your instructor know and they will walk you through these next steps.
   1. Create a feature branch in Visual Studio based off the master branch in VSTS (*remote origin).*
   2. Swap to the new feature branch and ensure you are “checked out” to that new feature branch.
   3. Make a code change to the project’s HTML page that you can visibly see an update to on the front-end web project.
   4. Commit the code in Visual Studio.
   5. Push the code from Visual Studio to VSTS.
   6. Create a Pull Request in VSTS
   7. Complete the Pull Request in VSTS to merge your code from your feature to your master branch.
3. Verify CI Build definition is running
4. Click on the Build tab.
5. You should see the Status of your build definition as **in progress**.
6. Click on the build # to see the status:



1. Once you get a successful build, continue to the next step.
2. Verify CD Release definition is running
3. Click on the Release tab.
4. You will see a blue in progress rectangle under the **Environments column.**
5. Click on the Release that is running.
6. Click Logs.
7. Once you get a successful release, continue to the next step.
8. Verify changes appear on App Service
9. In the Azure Portal, find your Web App resource.
10. Click on your Web App Resource.
11. On the Overview page you should see the URL for it.
12. Go to the URL and verify your changes are there.

Exercise 3 has been completed