**Application Lifecycle Management and Deploying Apps**

[Application Lifecycle Management and Deploying Apps 1](#_Toc40339106)

[Exercise 1: Initialize Azure DevOps 3](#_Toc40339107)

[Exercise 2: Build Export Pipeline 7](#_Toc40339108)

[Exercise 3: Test the Pipeline 17](#_Toc40339109)

[Exercise 4: Build Manage Solution and Publish Artifacts 22](#_Toc40339110)

[Exercise 5: Build a Release Pipeline 29](#_Toc40339111)

[Exercise 6: Share the apps with the users 39](#_Toc40339112)

[Scenario 39](#_Toc40339113)

[Exercise 7 (Optional): Test the application 42](#_Toc40339114)

[Scenario 42](#_Toc40339115)

[Exercise 8 (Optional): Setup the flow to use a service principal 47](#_Toc40339116)

[Scenario 47](#_Toc40339117)

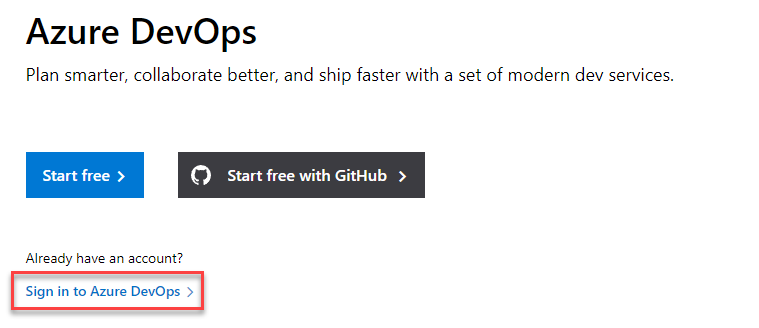
## **Exercise 1: Initialize Azure DevOps**

In this exercise, you will be signing up for an Azure DevOps account and configuring the PowerApps build tools for the account.

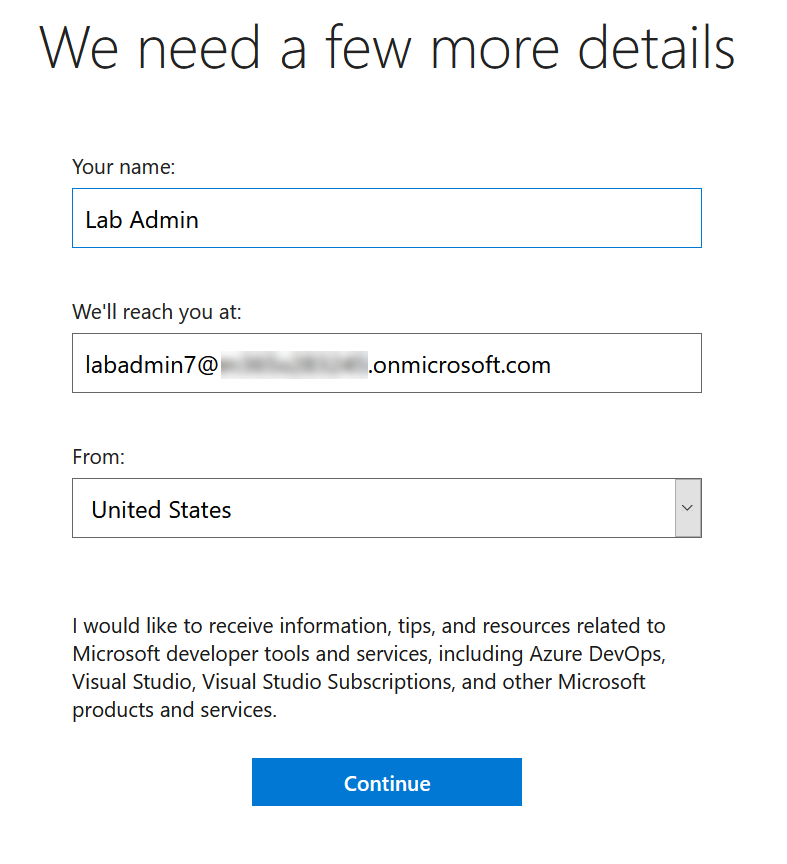
**Note:** If you already have Azure DevOps outside of this course and this environment, you CANNOT use that here. You will need to follow our instructions to sign up.

### **Task 1: Signup for Azure DevOps**

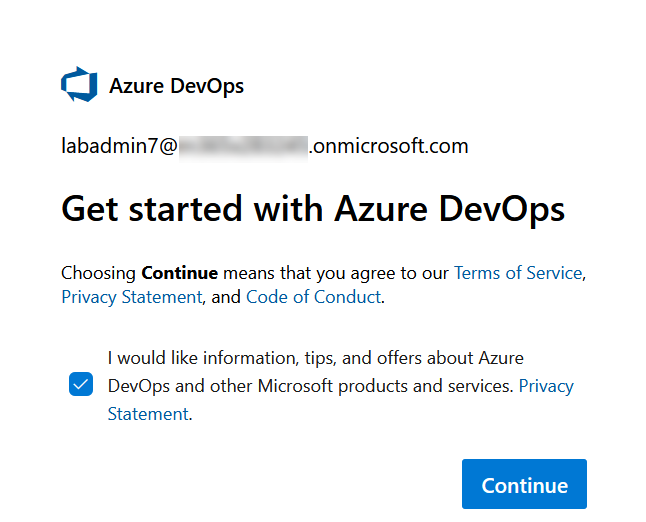
1. Navigate to <https://dev.azure.com> and click **Sign in to Azure DevOps**.   
     
   **Note:** Use the same account you have been using for the other labs.



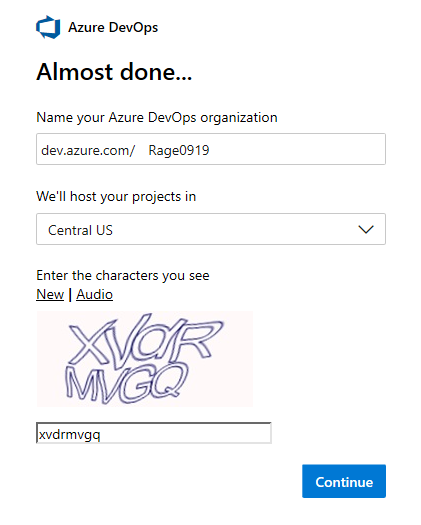
1. Provide your credentials and click **Next**.
2. Provide a **Password** and click **Sign in**.
3. Click **Continue**.



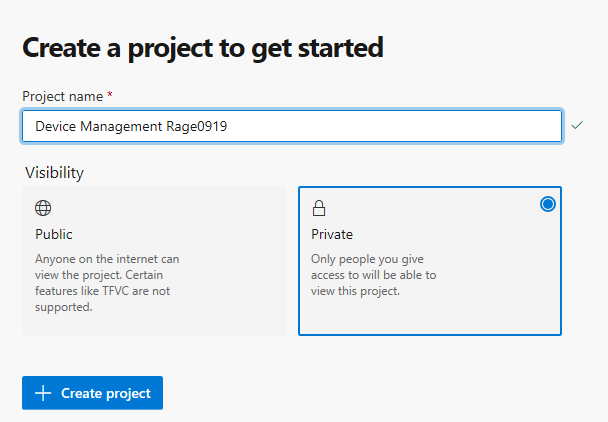
1. Click **Continue.**



1. If prompted, provide a unique **Azure DevOps Organization** name such as lastnameMMYY, select a location closest to your tenant, enter captcha and click **Continue**. Replace lastname with your last name, MM current month and YY current year.

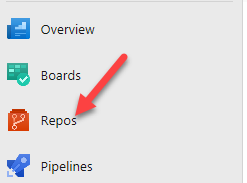


1. Enter **Device Management lastnameMMYY** for **Project Name** and click **Create**. Replace lastname with your Last name, MM with current month, and YY with current year.



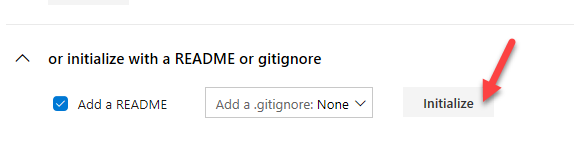
Projects are containers in Azure DevOps that track work items and source assets. When you set up the automation for the deployment tasks those will be pipelines built in the context of a project.

1. Select **Repos**.



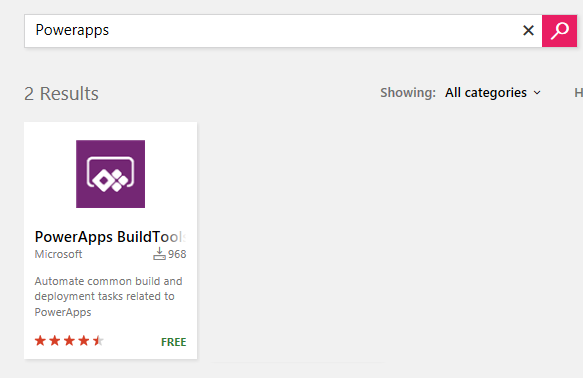
An Azure Repo is a source/version control container inside the Azure DevOps project used to track changes you make. You will be using it to store the solution files for the team building the Device Ordering app.

1. Check the **Add a Readme** checkbox and click **Initialize**.

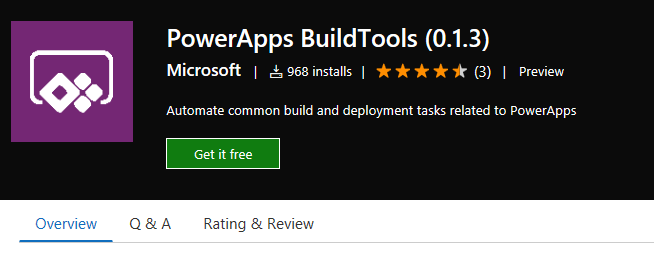


### **Task 2: Configure PowerApps Build Tasks**

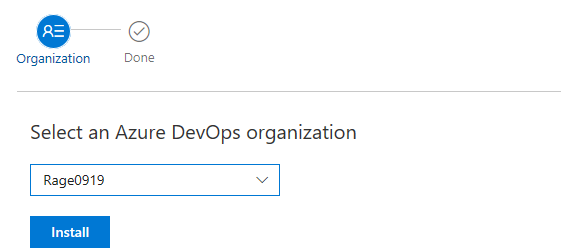
1. Navigate to <https://marketplace.visualstudio.com/azuredevops>
2. Search for **PowerApps**.
3. Select **PowerApps BuildTools**.



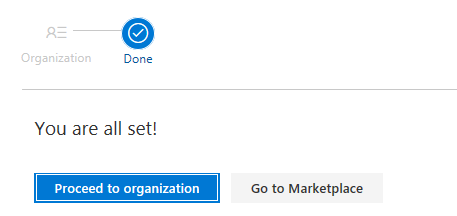
1. Click **Get it Free**.



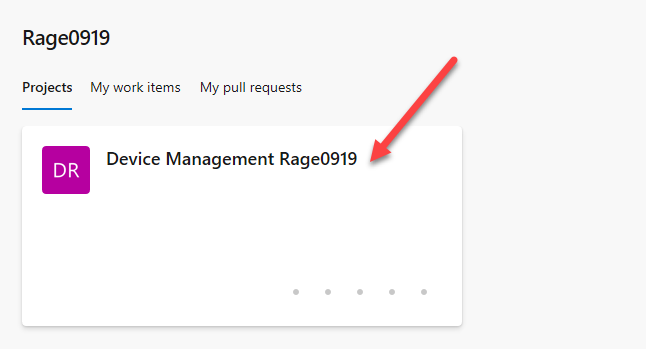
1. Select the **Azure DevOps** organization you created and click **Install**.

b

1. Click **Proceed to Organization**.



1. Click to open the **Device Management** project you created.

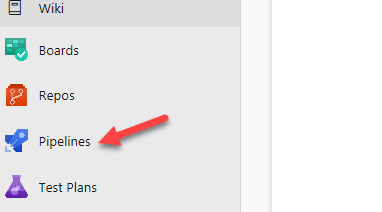


## **Exercise 2: Build Export Pipeline**

In this exercise, you will build an Azure DevOps pipeline that will export the solution from the development Common Data Service environment, unpack the solution file to individual files and then check those files into the repository. These solution files can be then used to re-create development environments or to promote the solution to test/production.

### **Task 1: Export Solution**

1. Create a Build Pipeline.
   1. Click to expand **Pipelines**.



* 1. Select **Pipelines**.

A screenshot of a cell phone

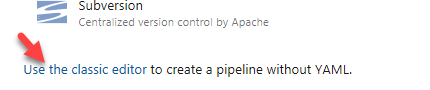
Description automatically generated

* 1. Click **Create Pipeline**.

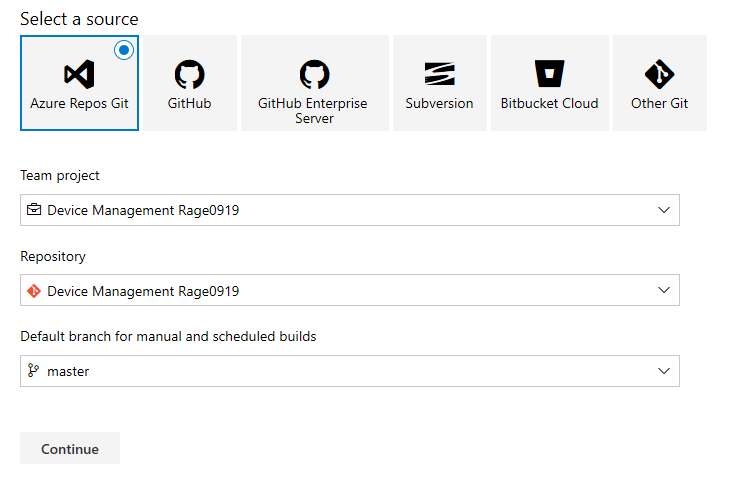
A screenshot of a cell phone

Description automatically generated

* 1. Click **Use the Classic Editor.**



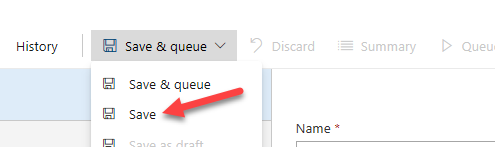
* 1. Don’t change the default values and click **Continue**.



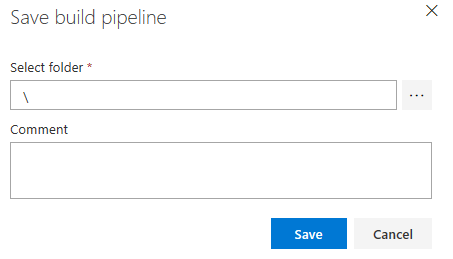
* 1. Select **Empty Job**.



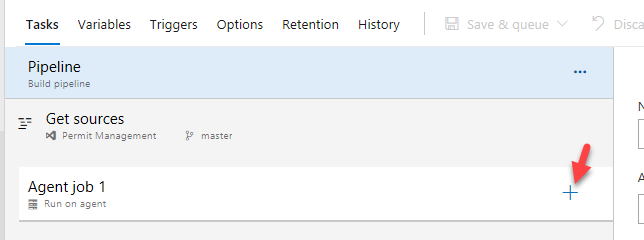
* 1. Click **Save and Queue** and select **Save**.



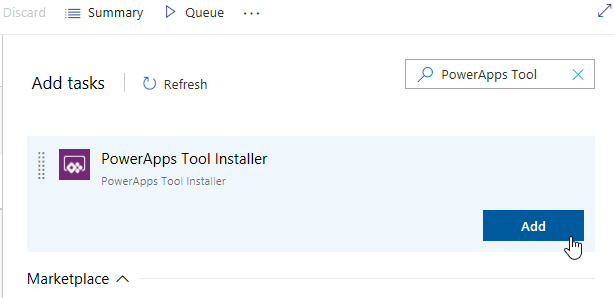
* 1. Click **Save**.



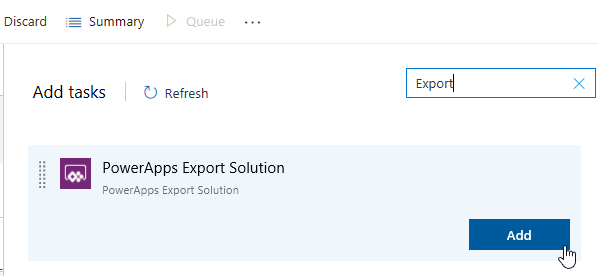
1. Add PowerApps Tool Installer task.  
   **Note:** The PowerApps Tool Installer needs to be run before any other PowerApps build tasks.
   1. Click **+ Add Task to Agent Job 1**.



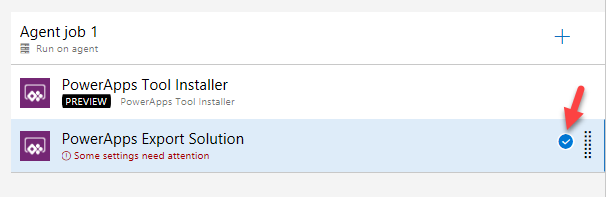
* 1. Search for **PowerApps Tool** hover over select **PowerApps Tool Installer** and click **Add**.



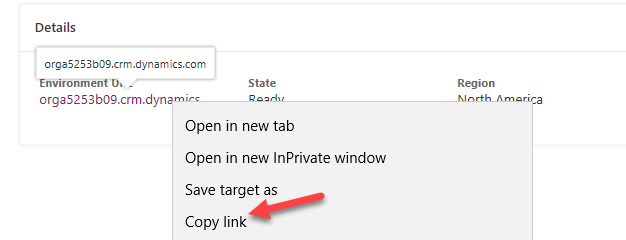
1. Add PowerApps Export Solution task.
   1. Search for **Export.**
   2. Hover over **PowerApps Export Solution** and click **Add**.



1. Open PowerApps Export Solution.
   1. Select the **PowerApps Export Solution** task.

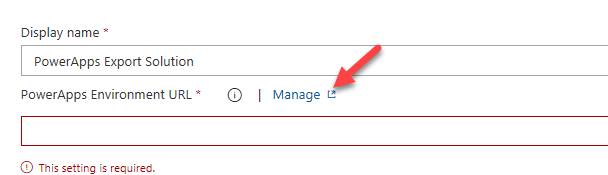


1. Get your development environment URL
   1. Start a new browser window or tab and navigate to [https://admin.powerplatform.microsoft.com](https://admin.powerplatform.microsoft.com/support)
   2. Select **Environments** and click to open the **Device Ordering Development** environment.
   3. Copy the **Environment URL** and keep it in your clipboard or you can keep this URL on a notepad.

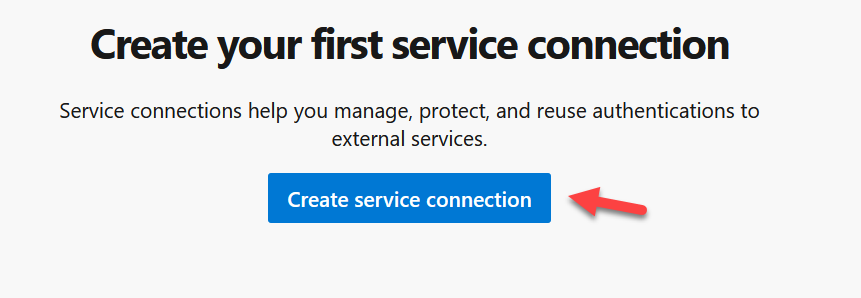


* 1. Close the **Power Platform Admin** browser window or tab.

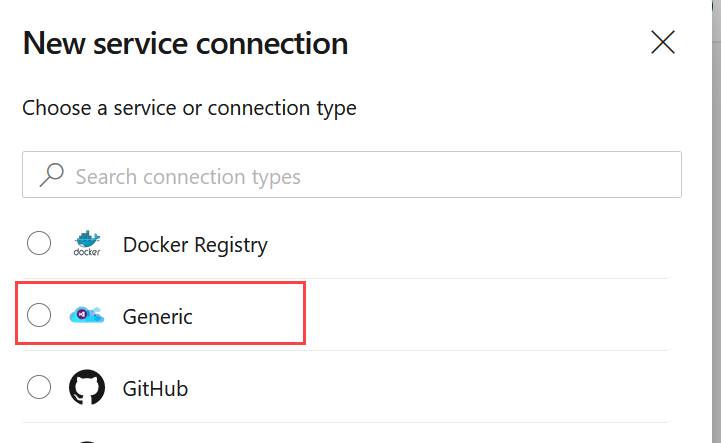
1. Create a **Generic Service Connection.** Service Connections are how the build tasks know what environment URL and user credentials to use to access the Common Data Service environments.
   1. Go back to the **Pipeline**.
   2. Make sure you still have the **PowerApps Export Solution** task selected.
   3. Click **Manage**.



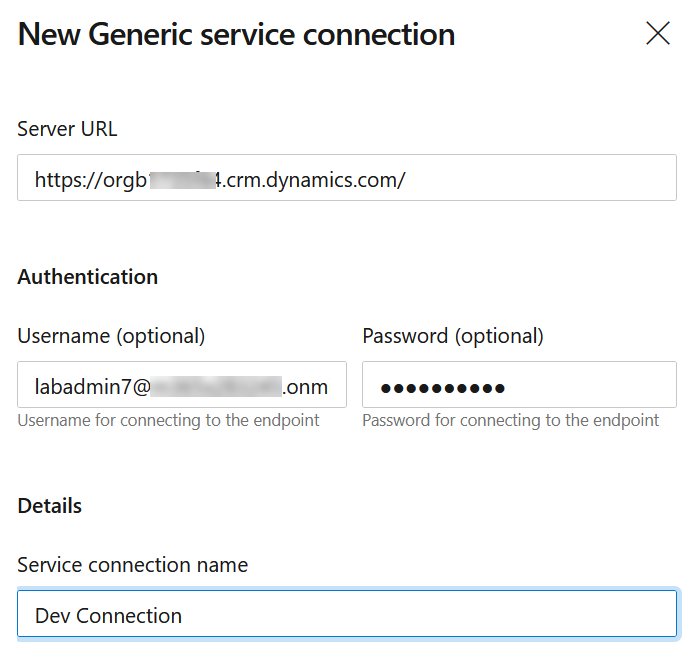
* 1. Click Create Service Connection.



* 1. Select **Generic** and then click Next.

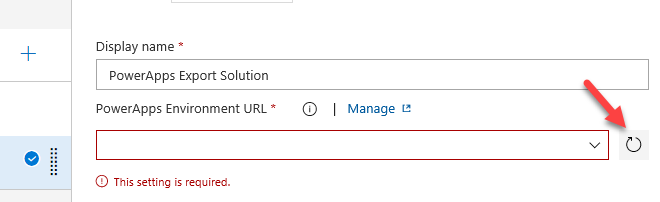


* 1. Provide a **Connection Name**, paste the **Environment URL** you copied, provide your credentials and click Save.

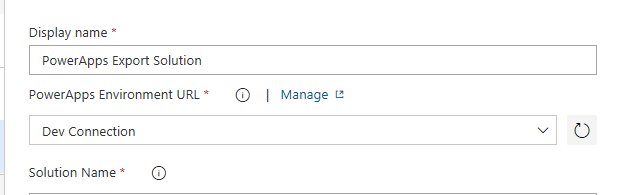


* 1. Close the **Service Connections** browser window or tab.

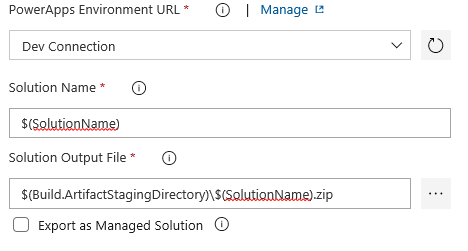
1. Select the Generic Service Connection you created as the Power Apps Environment URL.
   1. Go back to the **Build Pipeline** tasks and make sure you still have PowerApps Export Solution task selected.
   2. Locate the **PowerApps Environment URL** field and click **Refresh**.



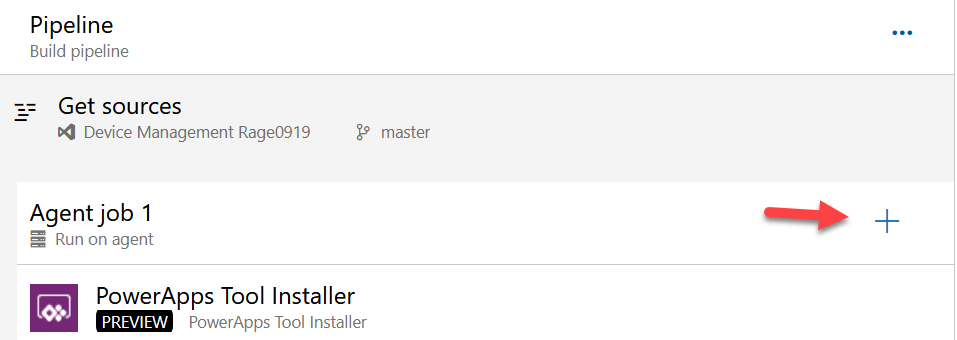
* 1. Select the **Generic Service Connection** you just created named Dev Connection.



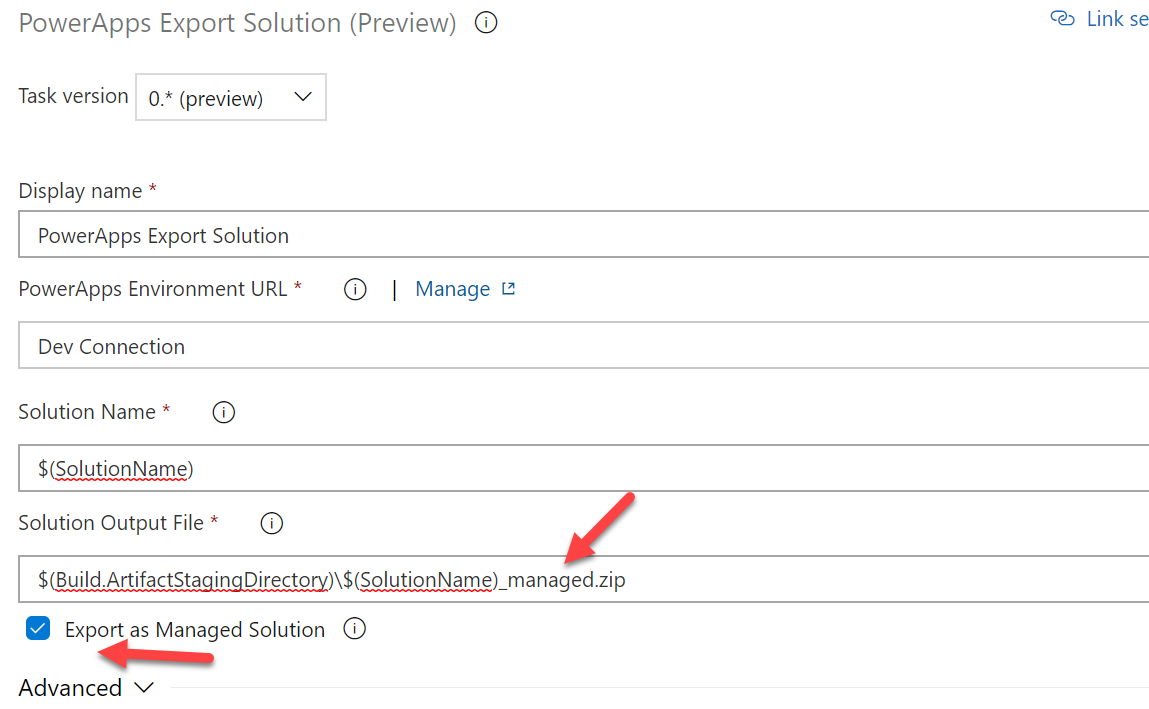
* 1. Enter **$(SolutionName)** for **Solution Name**, **$(Build.ArtifactStagingDirectory)\$(SolutionName).zip** for **Solution Output File**.



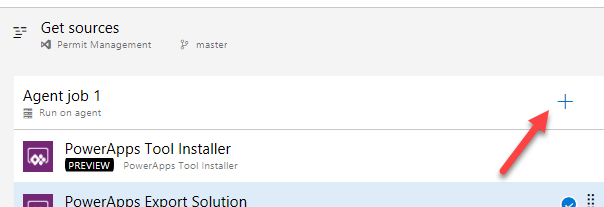
* 1. Click +Add Task and add another Export Solution task.



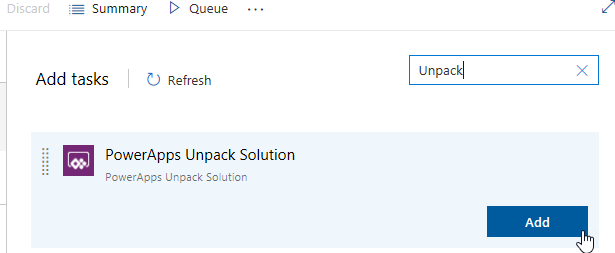
* 1. Select **Dev Connection** for the PowerApps Environment URL.
  2. Check the Export as Managed Solution.
  3. Enter **$(SolutionName)** for **Solution Name**, **$(Build.ArtifactStagingDirectory)\$(SolutionName)\_managed.zip** for **Solution Output File**



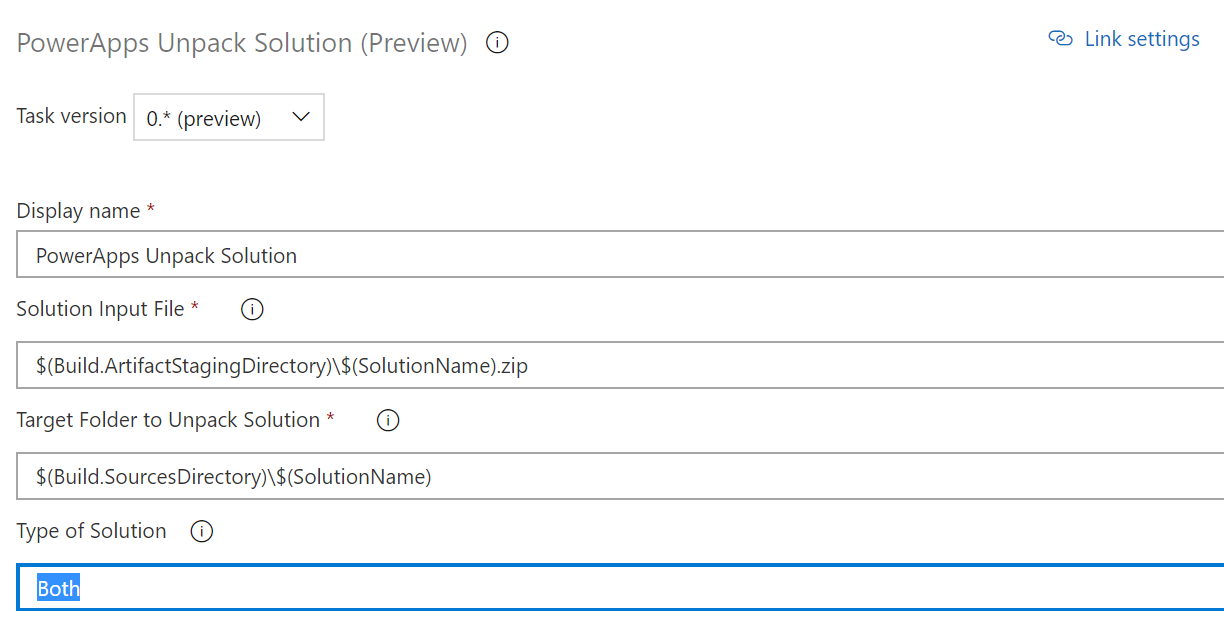
1. Add an Unpack task. This task will take the solution zip file and expand it into a file for each solution component.
   1. Click **+ Add Task**.



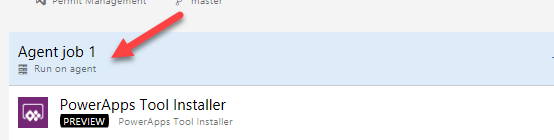
* 1. Search for **Unpack**.
  2. Hover over **PowerApps Unpack Solution** and click **Add**.



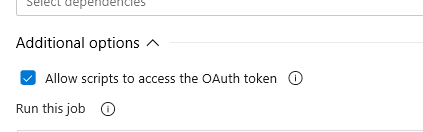
1. Provide Unpack settings information.
   1. Select the **Unpack** task.
   2. Enter **$(Build.ArtifactStagingDirectory)\$(SolutionName).zip** for **Solution Input** **File**, **$(Build.SourcesDirectory)\$(SolutionName)** for **Target Folder**.
   3. Choose **Both** for Type of Solution.



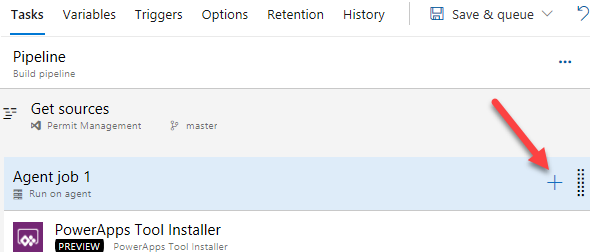
1. Allow scripts to access the OAuth Token. This will allow the commands you will add to check in files to the Azure DevOps repo to work.
   1. Select **Agent Job 1**.



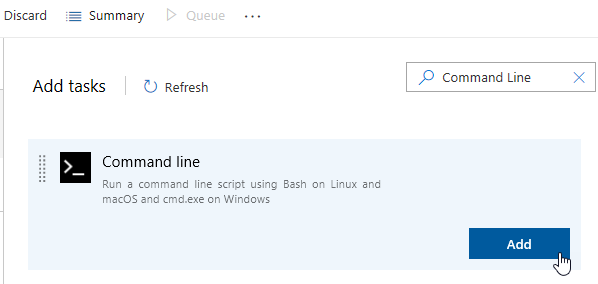
* 1. Scroll down and check the **Allow Scripts to Access the OAuth Token** checkbox.



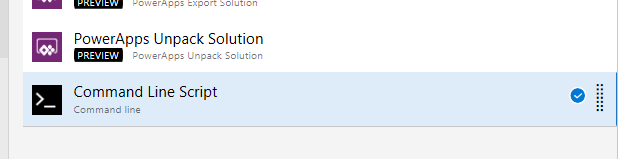
1. Add Command Line task.
   1. Click **+ Add a Task**.



* 1. Search for **Command Line**.
  2. Hover over **Command Line** and click **Add**.



1. Add Scripts to the Command Line task. This task will be used to check in the solution file changes to the repo.
   1. Select the Command Line task.



* 1. Paste the script below in the **Script** text area. Replace **user@myorg.onmicrosoft.com** with your username.

echo commit all changes

git config user.email "user@myorg.onmicrosoft.com"

git config user.name "Automatic Build"

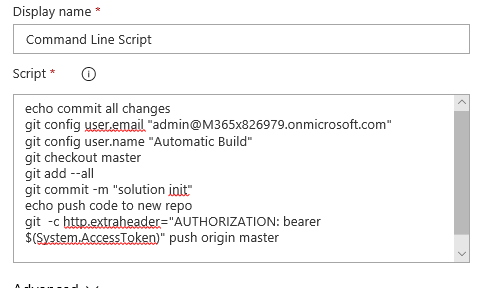
git checkout master

git add --all

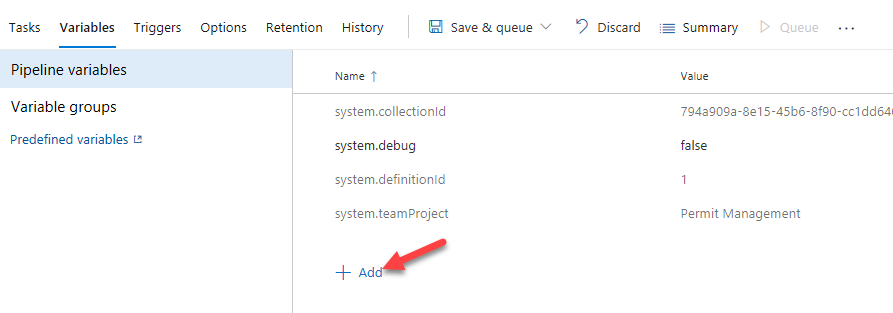
git commit -m "solution updates"

echo push code to new repo

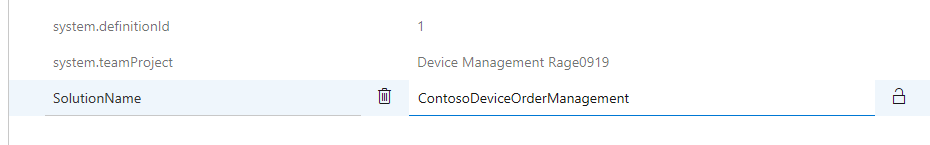
git -c http.extraheader="AUTHORIZATION: bearer $(System.AccessToken)" push origin master



1. Add Solution Name variable.
   1. Select the Variables tab.
   2. Click **+ Add.**



* 1. Enter **SolutionName** for **Name** and **ContosoDeviceOrderManagement** for **Value**. Confirm that there is no extra spaces at the end of the value after pasting.



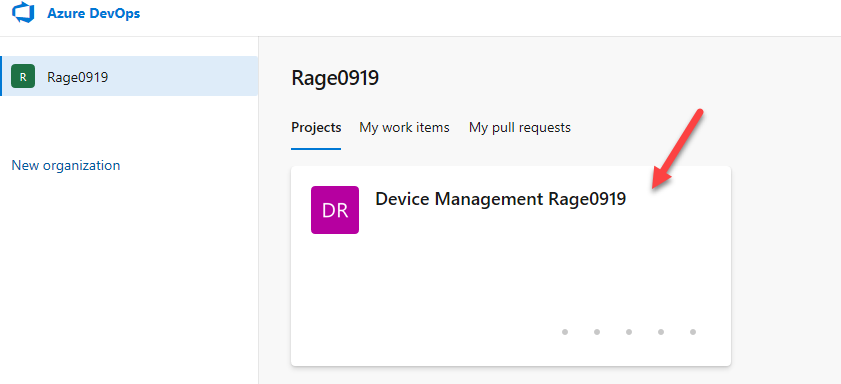
* 1. Click **Save and Queue** and select **Save**.
  2. Click **Save** again.

## **Exercise 3: Test the Pipeline**

In this exercise, you will test the build pipeline you created.

### **Task 1: Run the Pipeline**

1. Open the build pipeline.
   1. Navigate to <https://dev.azure.com/> and click to open the **Device Management** project.



* 1. Select **Pipelines | Pipelines**.

A screenshot of a cell phone

Description automatically generated

* 1. Click **Device Management-youruser-CI**.

A screenshot of a cell phone

Description automatically generated

* 1. Click **Run Pipeline**A screenshot of a cell phone

     Description automatically generated
  2. Select **Run** again and wait.  
     A screenshot of a cell phone

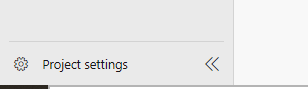
     Description automatically generated
  3. Refresh the screen until your Agent job 1 has a Status of **Running**  
     A screenshot of a social media post

     Description automatically generated
  4. Click on Agent job 1 to open the job
  5. The Build will run up to the Command Line Script and fail because we need to adjust permissions. The first run causes a project specific build service user to be added that we will in the next steps adjust the permissions to fix this problem.

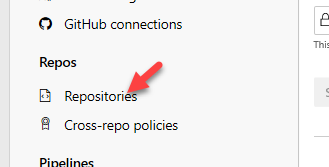
A screenshot of a cell phone

Description automatically generated

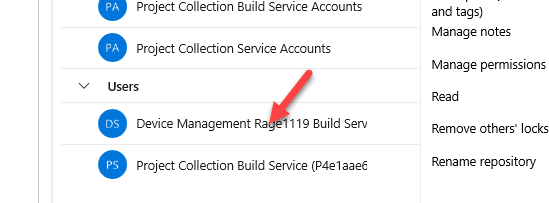
* 1. Click **Project Settings** in the lower left corner



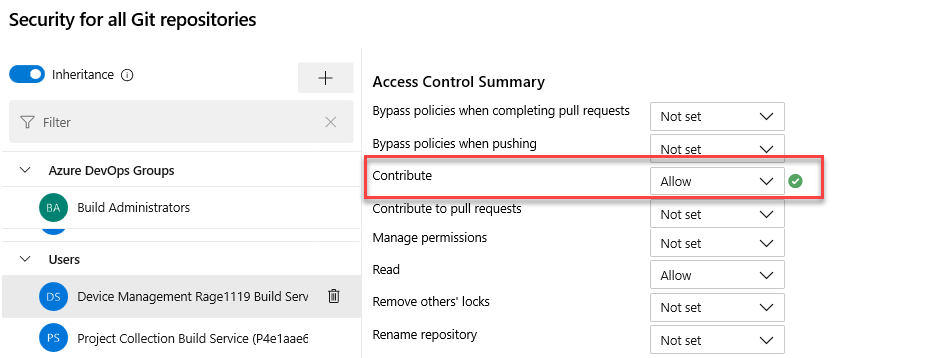
* 1. Select **Repositories**.



* 1. In the Users section, Select the **Device Management Build Service** user.



* 1. In the **Access Control Summary**, locate **Contribute** and change it to **Allow**.



* 1. Select **Pipelines** | **Pipelines** in the left navigation
  2. Click **Queue** on your build pipeline
  3. Click **Run** and wait
  4. The build should now run successfully

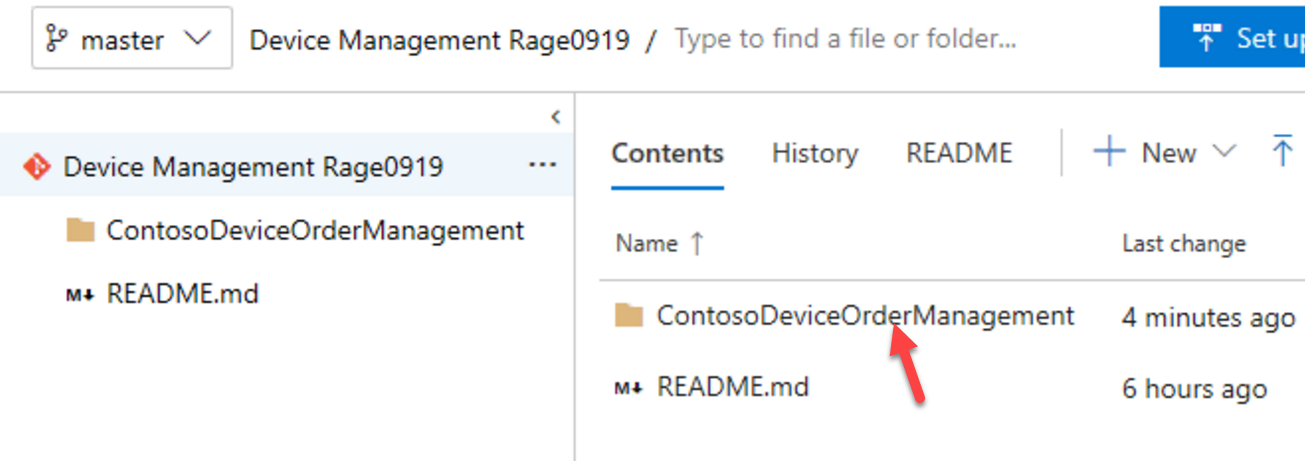
A screenshot of a cell phone

Description automatically generated

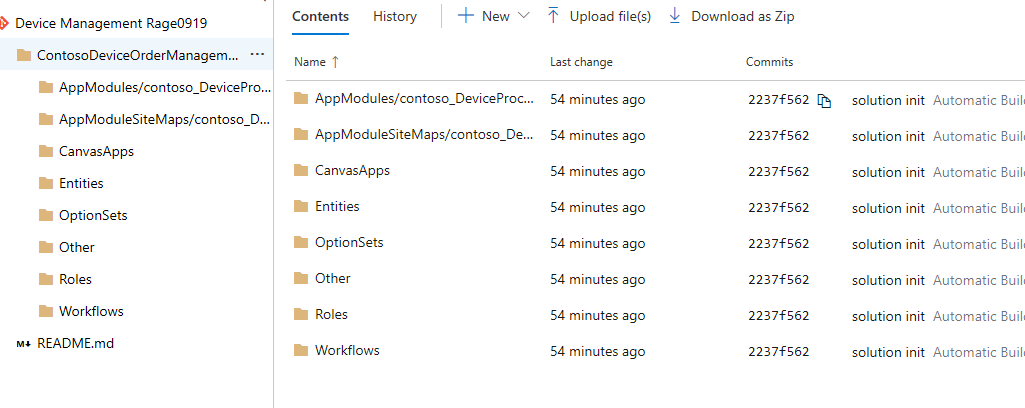
1. Review the Repository.
   1. Select Repos.



* 1. You should see **ContosoDeviceOrderManagement** folder. Click to open the folder.



* 1. The content of the folder should look like the image below.



* 1. You may examine the content of the folders.

## **Exercise 4: Build Manage Solution and Publish Artifacts**

In this exercise, you will take the solution files checked into the repo in the prior steps and re-package them into a managed solution file. This solution file will be published as a build artifact so it can be used in the release pipeline we are going to use to publish to test and production.

In a real project this is where you could add steps to import the solution into a build Common Data Service environment to check for missing dependencies. You could also add build tasks to run tests against your solution as well as run PowerApps Solution Checker to detect problems. In this lab exercise we will skip those extra steps to ensure you have enough time to complete the lab.

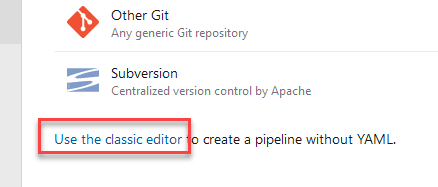
### **Task 1: Build the Managed Solution**

1. Select **Pipelines**, click **New Pipeline**.

A screenshot of a cell phone

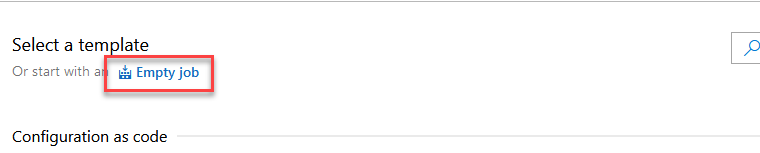
Description automatically generated

1. Click **Use Classic Editor**.

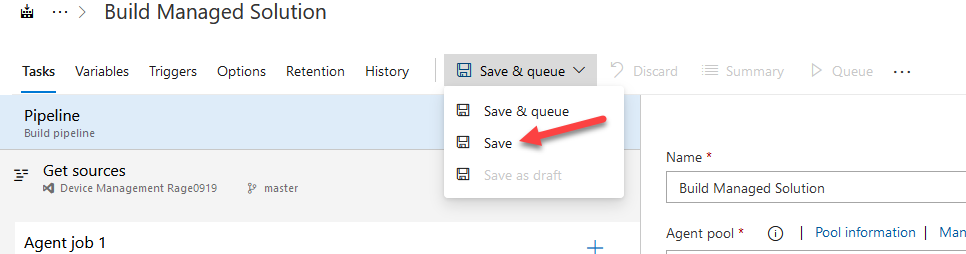


1. Click **Continue**.  
   A screenshot of a cell phone

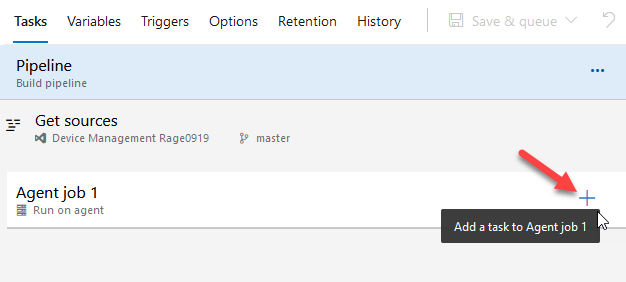
   Description automatically generated
2. Click **Empty Job.**



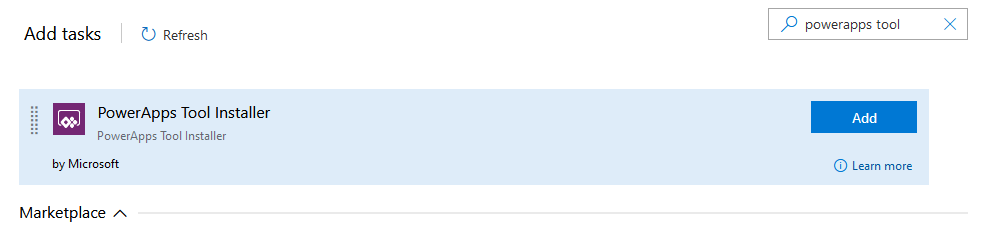
1. Enter **Build Managed Solution** for **Name**, click **Save and Queue** and select **Save**.



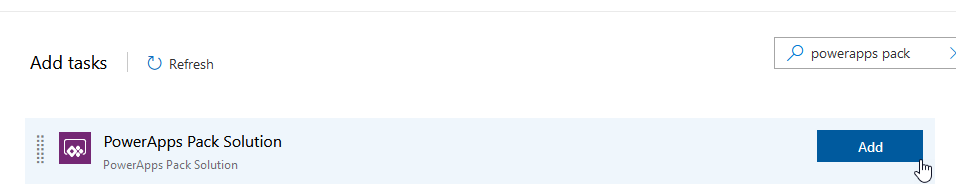
1. Click **Save** again.
2. Click **Add a Task to Agent Job 1**.



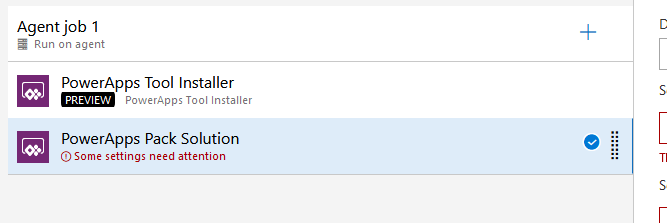
1. Search for **PowerApps Tool**, hover over **PowerApps Tool Installer** and click **Add**.



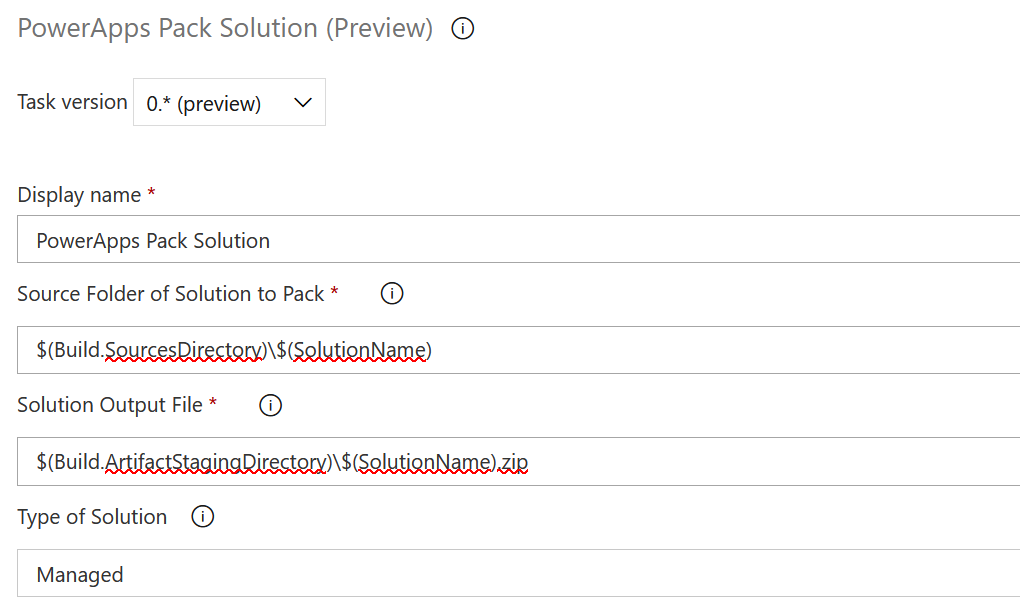
1. Search for **PowerApps Pack**, hover over **PowerApps Pack Solution** and click **Add**.



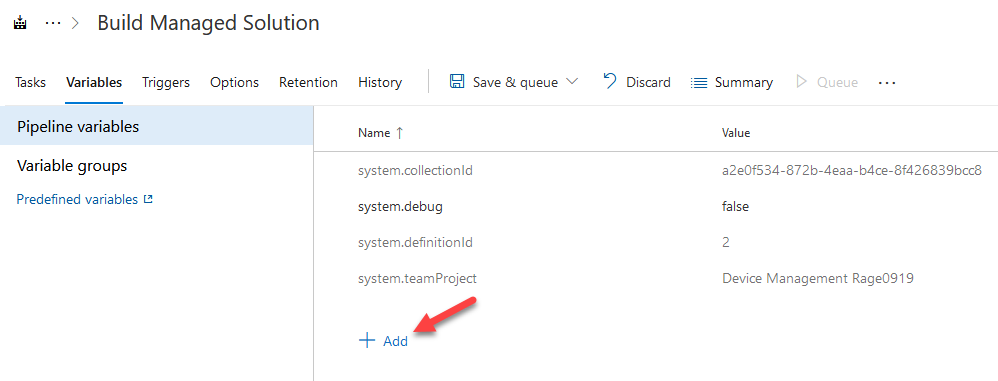
1. Select the **PowerApps Pack Solution** task.



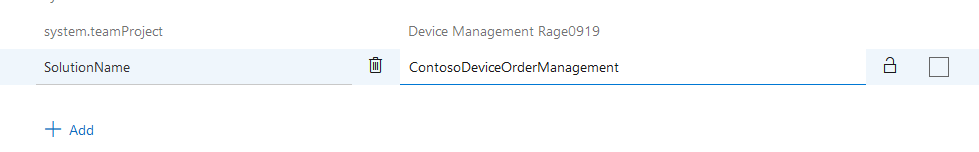
1. Enter **$(Build.SourcesDirectory)\$(SolutionName)** for **Source Folder of Solution to Pack**, enter **$(Build.ArtifactStagingDirectory)\$(SolutionName).zip** for **Solution Output Folder**.
2. Change the solution type to Managed



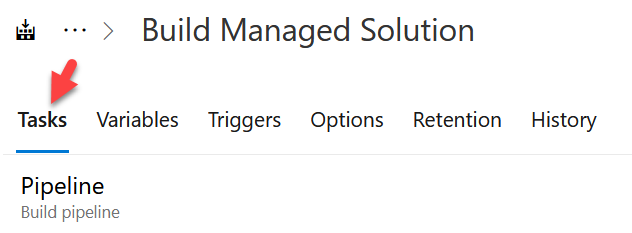
1. Select the **Variables** tab and click **+ Add**.



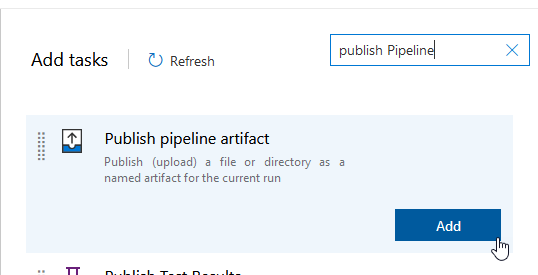
1. Enter **SolutionName** for **Name** and **ContosoDeviceOrderManagement** for **Value**.



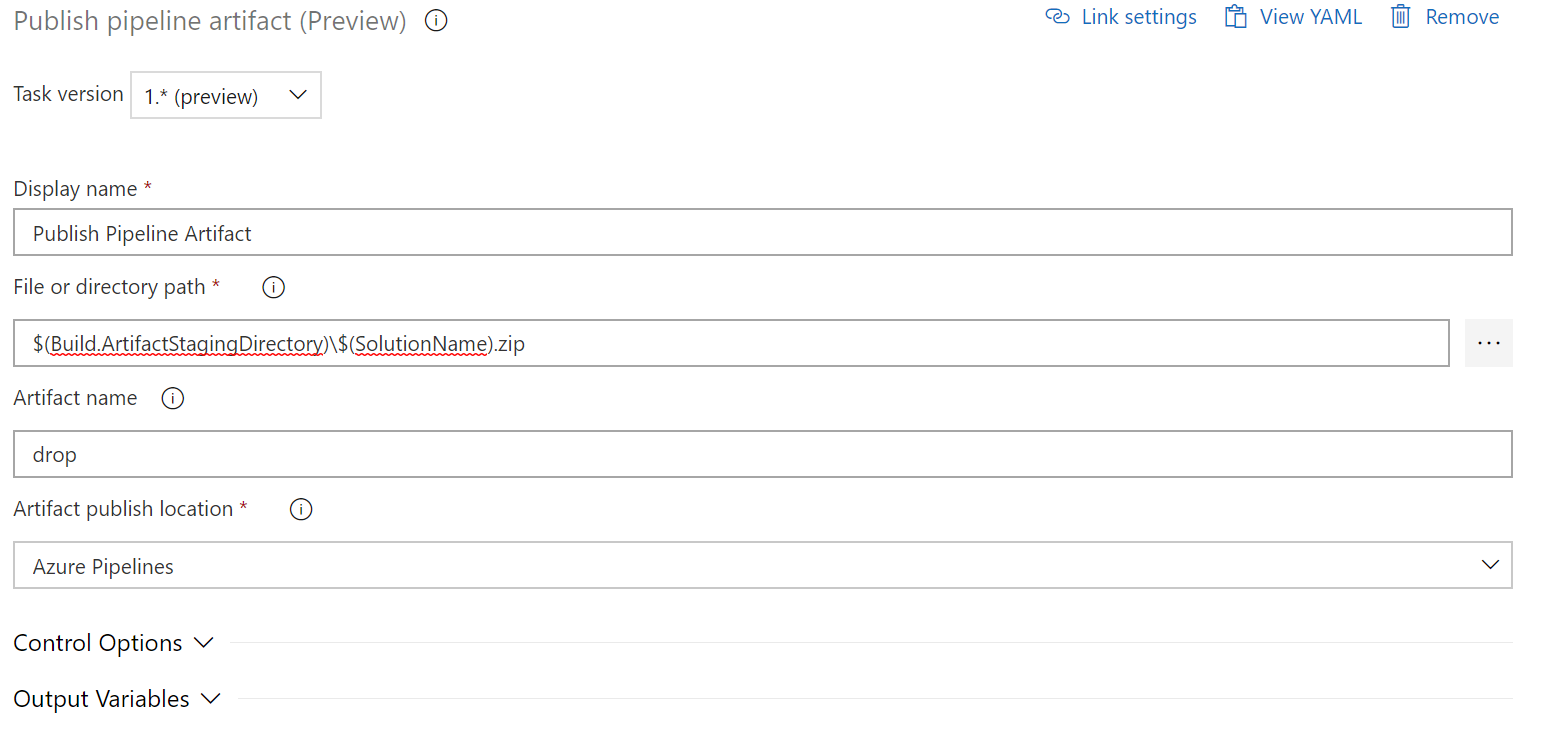
1. Navigate back to **Tasks.**



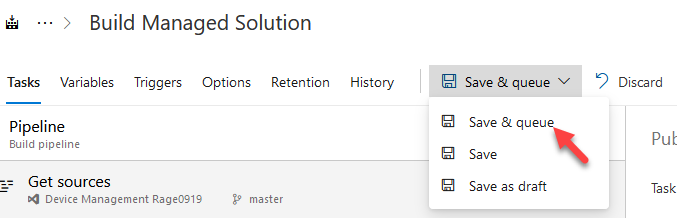
1. Click **Add a Task**.
2. Search **Publish Pipeline**, hover over **Publish Pipeline Artifact** and click **Add**. Publishing the solution as an artifact will make it available for the release pipeline you will build.



1. Select the Publish Pipeline Artifact task.
2. Enter **$(Build.ArtifactStagingDirectory)\$(SolutionName).zip** for **File or Directory Path** and enter drop for **Artifact Name**.



1. Click **Save and Queue** and select **Save and Queue**.



1. Click **Save and Run**.
2. Refresh the page until Agent Job 1 has a status of **Running**  
   A screenshot of a social media post

   Description automatically generated
3. Click on Agent job 1 and wait for the run to complete. Build tasks should run and succeed.

A screenshot of a cell phone

Description automatically generated

1. Click on **1 artifact produced**

A screenshot of a cell phone

Description automatically generated

1. Expand the folder and you should see the managed solution.

A screenshot of a cell phone

Description automatically generated

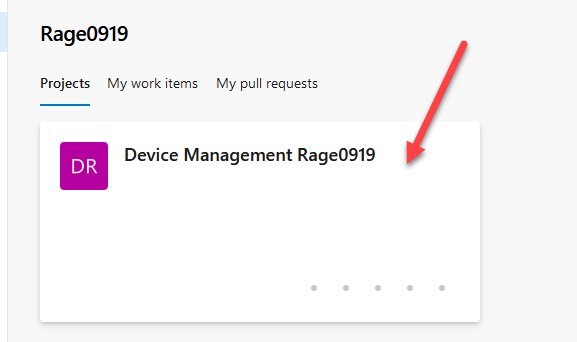
## **Exercise 5: Build a Release Pipeline**

In this exercise, you will build your release pipeline. The release pipeline is intended to take the output from the build pipeline and coordinate deploys to one or more release environments. A common release pipeline might deploy to dev -> test -> user acceptance -> production. Release pipelines can have approval requirements between each environment.

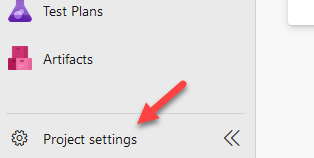
For the purposes of this lab we are only going to deploy to one test environment. For the purposes of this lab, we will be using the Central Apps Test trial environment we created in the previous lab. In real use, you would deploy to both Test and Production and would use environments of type production instead of trial.

### **Task 1: Create Connection and Release to Prod**

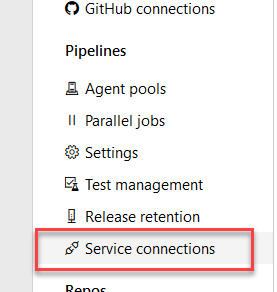
1. Go back to <https://dev.azure.com/>
2. Click to open the **Device Management** project.



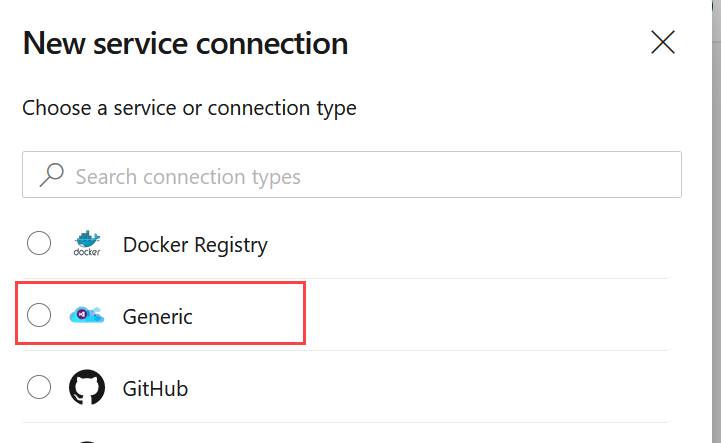
1. Click **Project Settings.**



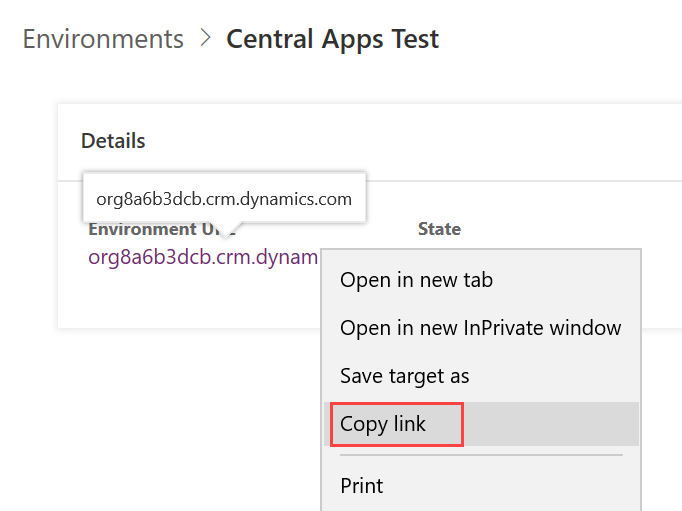
1. Go to the **Pipelines** area and select **Service Connections**.



1. Click **New Service Connection** and select **Generic**.



1. In a new tab, navigate to <https://admin.powerplatform.microsoft.com>
2. Select Environments and locate your Central Apps Test environment you created in the prior lab.
3. Click on the environment name to show the details, and then copy the URL from the detail screen.

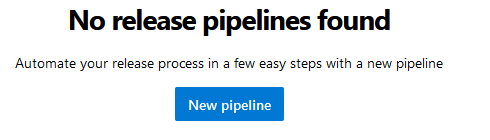


1. Enter **Test** for **Connection Name**, paste the **Environment URL** copied in the previous step, provide your credentials and click **OK**.
2. Select **Pipelines** -> **Releases**.

A screenshot of a cell phone

Description automatically generated

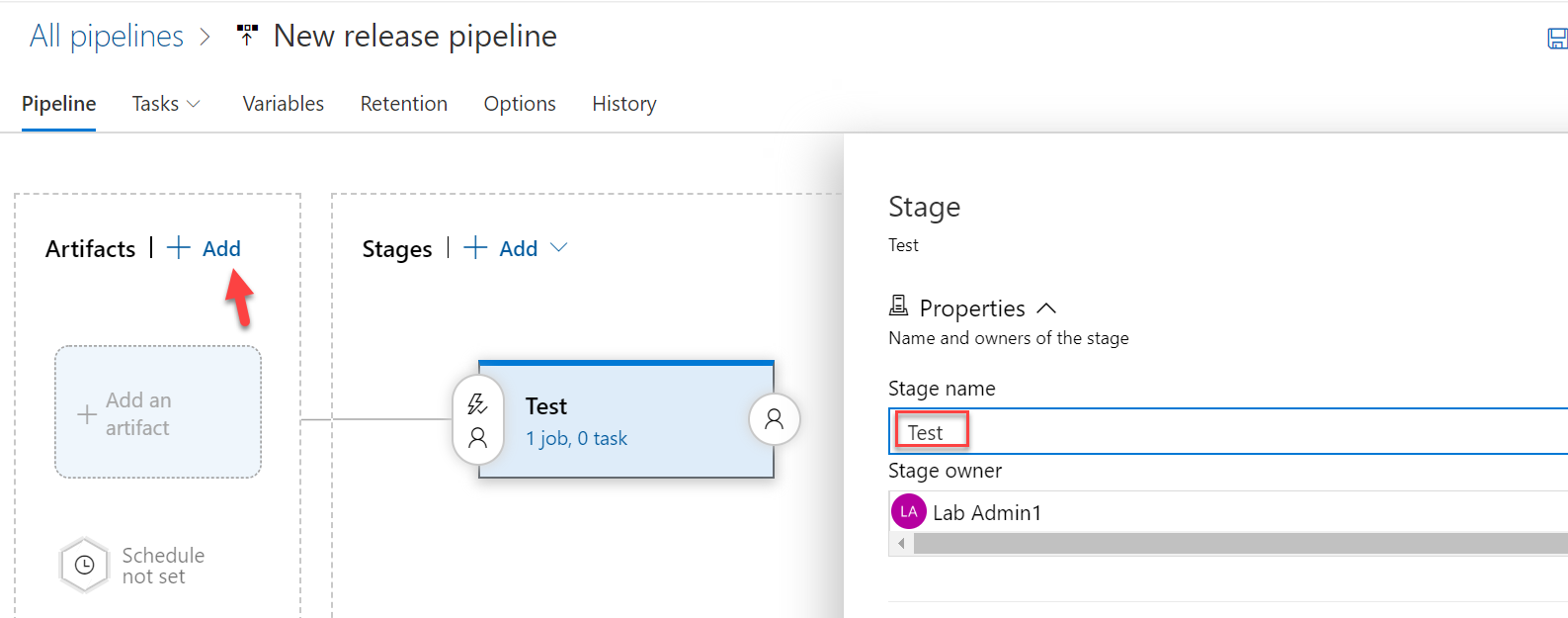
1. Click **New Pipeline**.



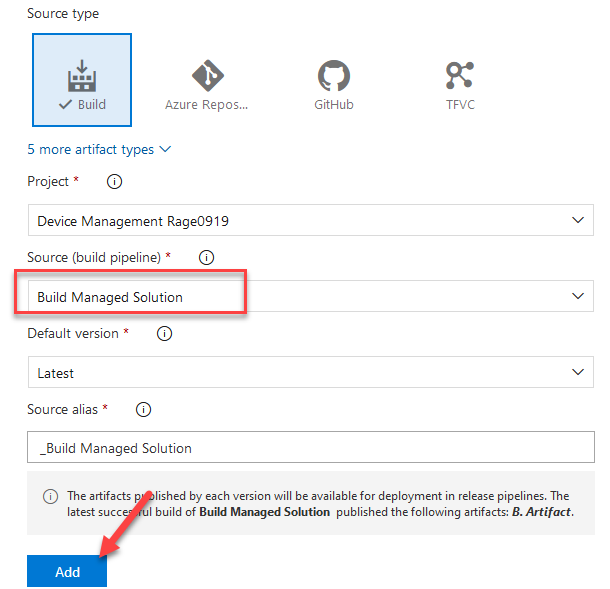
1. Select **Empty Job**.



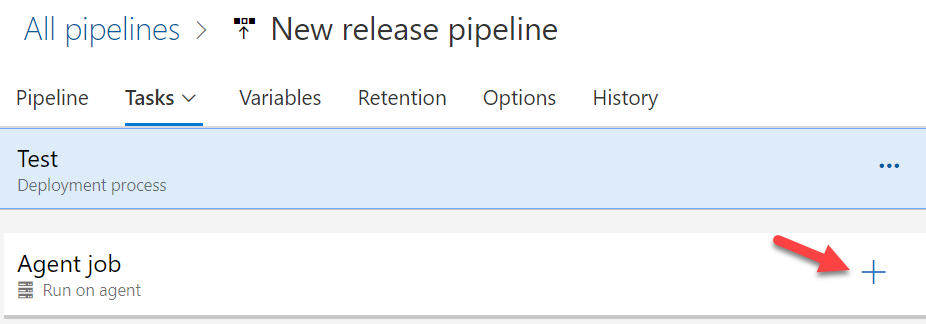
1. Enter **Test** for **Stage Name** and click **+ Add Artifacts**.



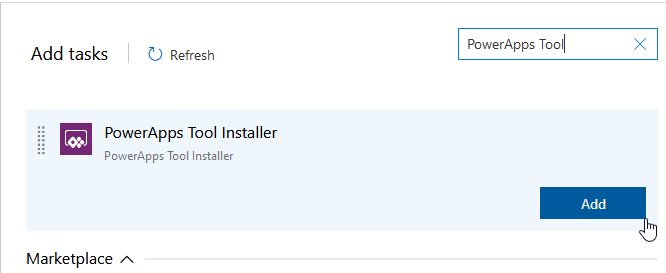
1. Select **Build Managed Solution** for **Source** and click **Add**.



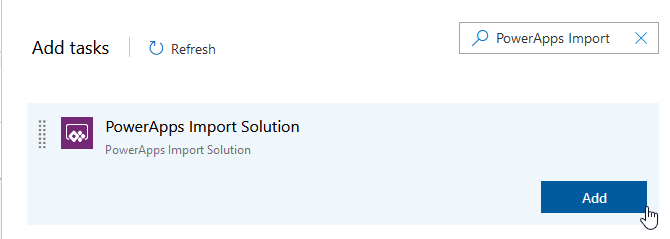
1. Select the **Tasks** tab and click **+ Add Task**.



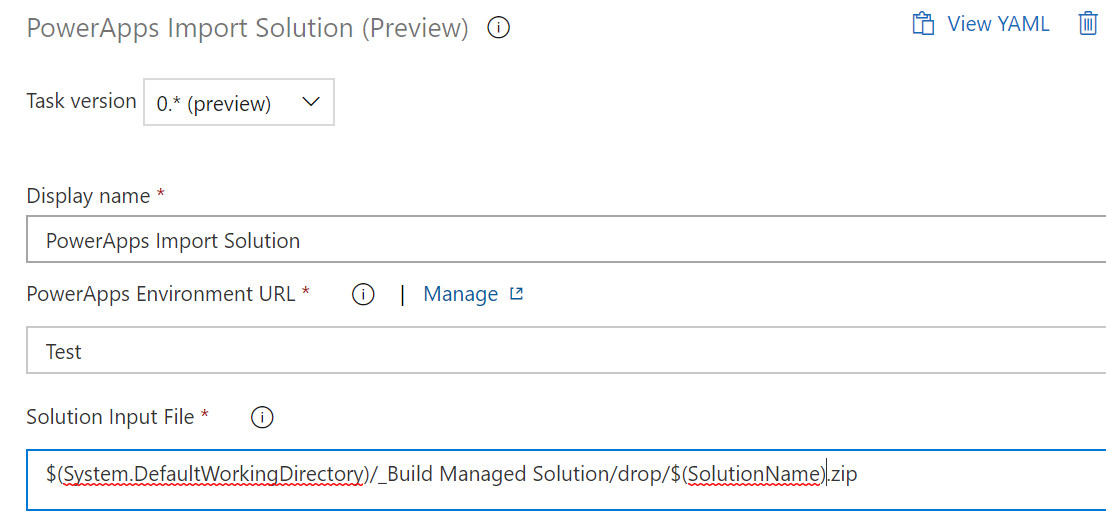
1. Search for **PowerApps Tool**, hover over **PowerApps Tool Installer** and click **Add**.



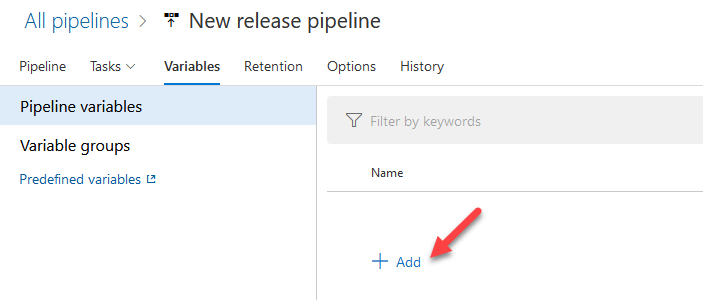
1. Search for **PowerApps Import**, hover over **PowerApps Import Solution** and click **Add**.



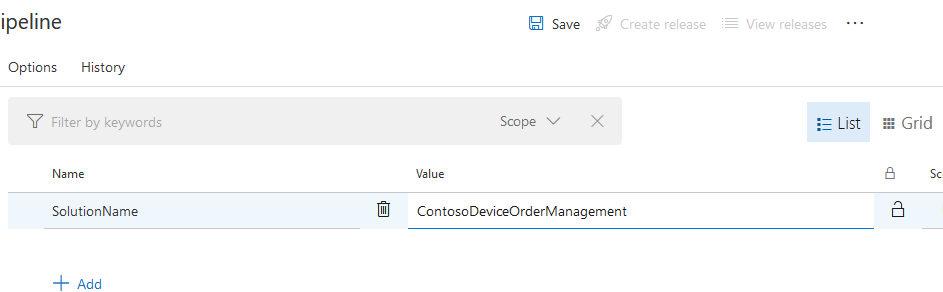
1. Select the **PowerApps Import Solution** task.
2. Select the **Test** connection for Environment and enter **$(System.DefaultWorkingDirectory)/\_Build Managed Solution/drop/$(SolutionName).zip** for **Solution Input File**.



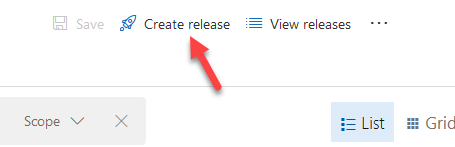
1. Select the **Variables** tab and click **+ Add**.



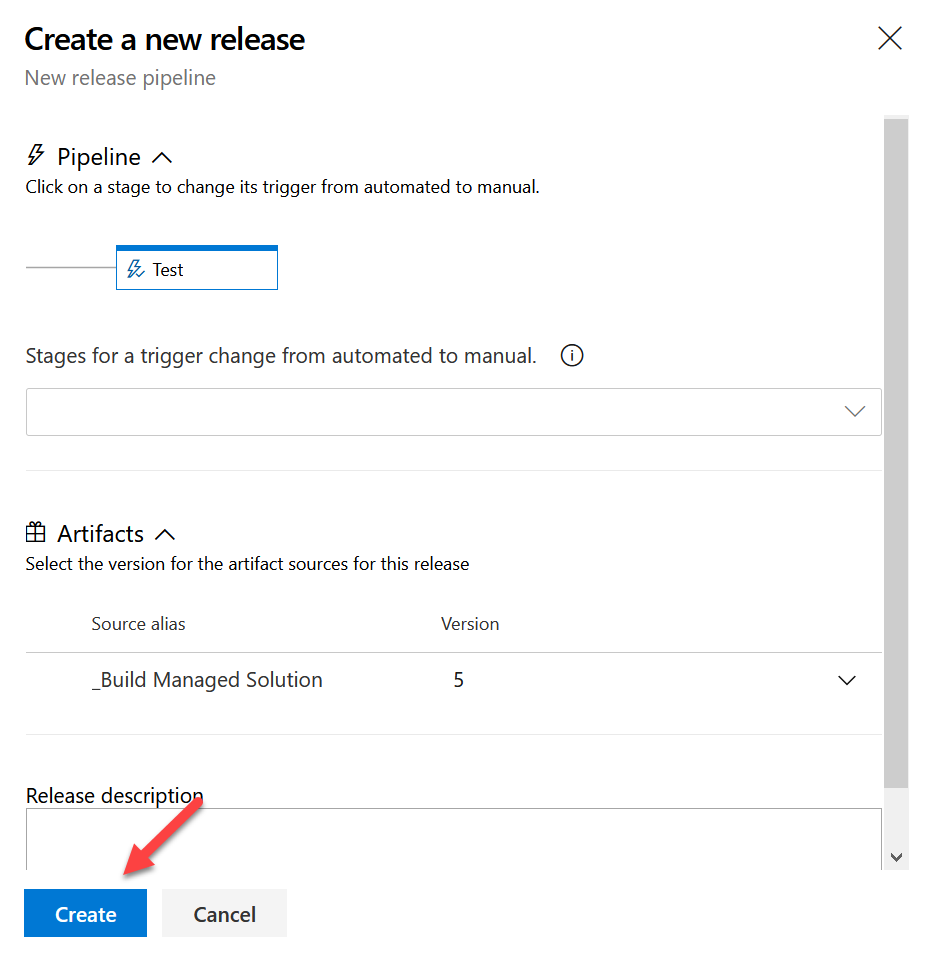
1. Enter **SolutionName** for **Name**, ContosoDeviceOrderManagement for **Value**, and click **Save**.



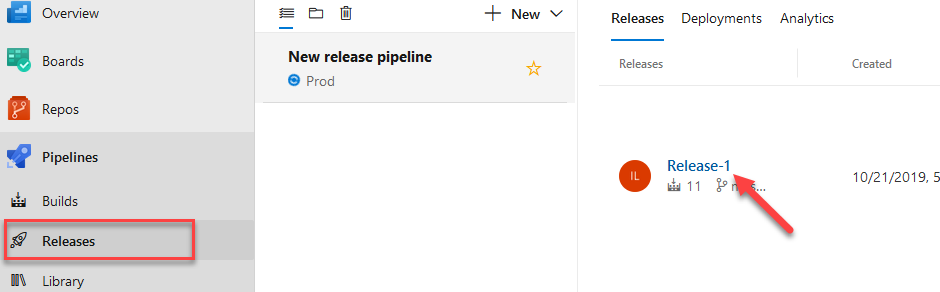
1. Click **OK**.
2. Click **Create Release**.



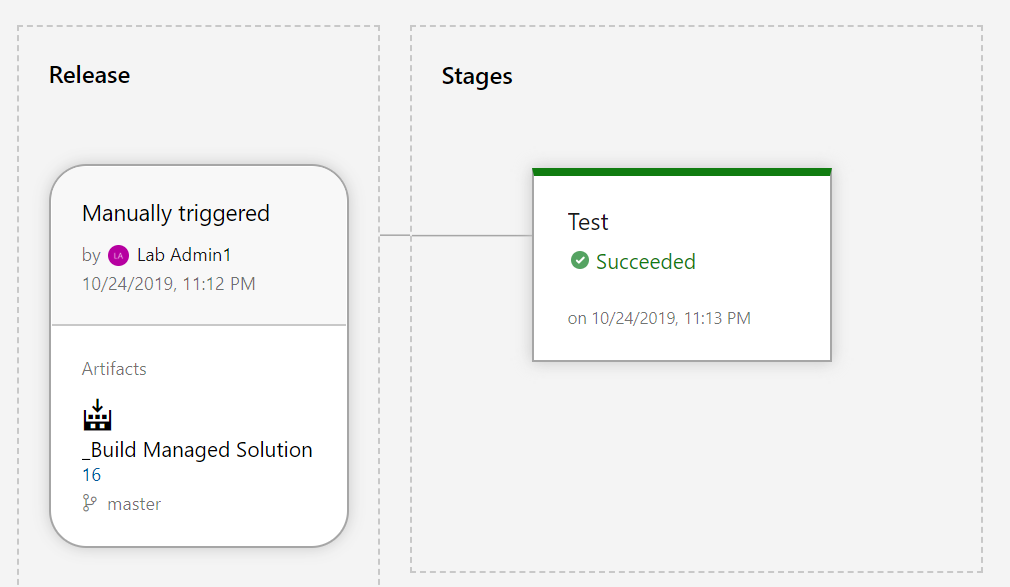
1. Click **Create**.



1. Select Releases and click to open the release.



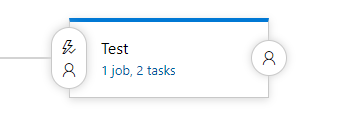
1. Wait for the release tasks to complete. The release pipeline should succeed.

  
  
If your Release has succeeded, scroll to Step 27  
If your Release has failed, try:

A common error here is the import fails because the Solution file is not found – one of the reasons for this could be that you have named your pipeline different to what’s written in the lab.  
A screenshot of a cell phone

Description automatically generated  
  
A fix for this is to manually select the solution file for the import. To do click on **Edit 🡪 Edit Pipeline**A screenshot of a cell phone

Description automatically generated

Select the **Test Task**  
  
Select **PowerApps Import Solution**Select the … next to **Solution Input File**A screenshot of a computer

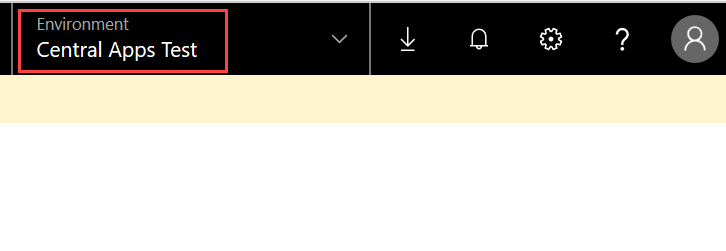
Description automatically generated

Select the .zip file in the drop folder and confirm with OK  
A screenshot of a social media post

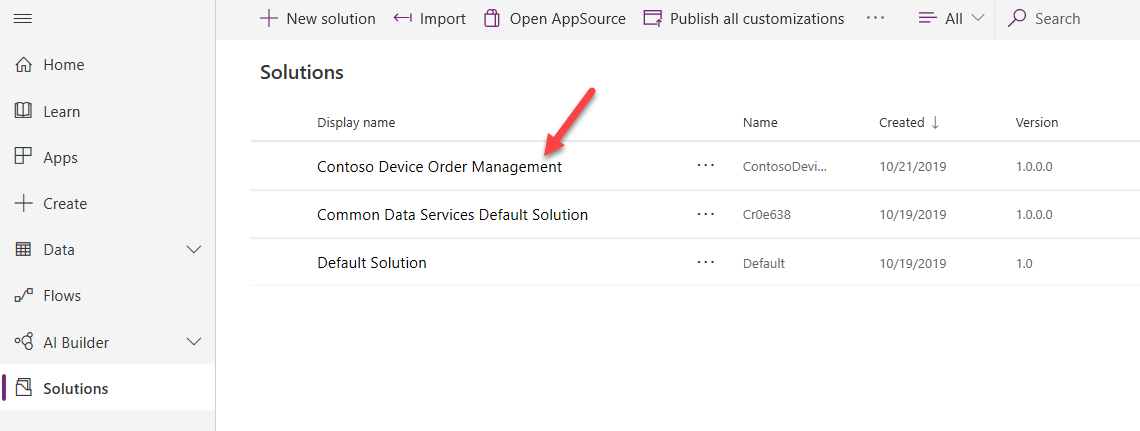
Description automatically generated

Go back to Releases and click **Create Release and Create**

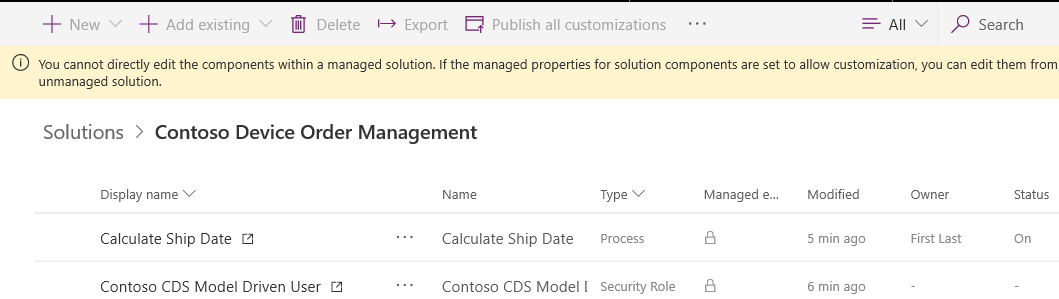
1. Navigate to <https://make.powerapps.com> and select the **Test** environment Central Apps Test.



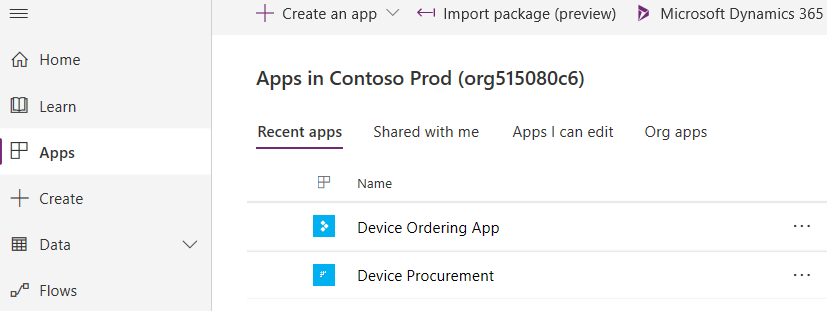
1. Select **Solutions**. You should see the managed solution you published from then release pipeline.
2. Click to open the Contoso Device Order Management solution.



1. You should get the **You cannot edit managed** solution message.



1. Select Apps. You should see **Device Ordering App** Canvas application and **Device Procurement** Model-Driven-Application.



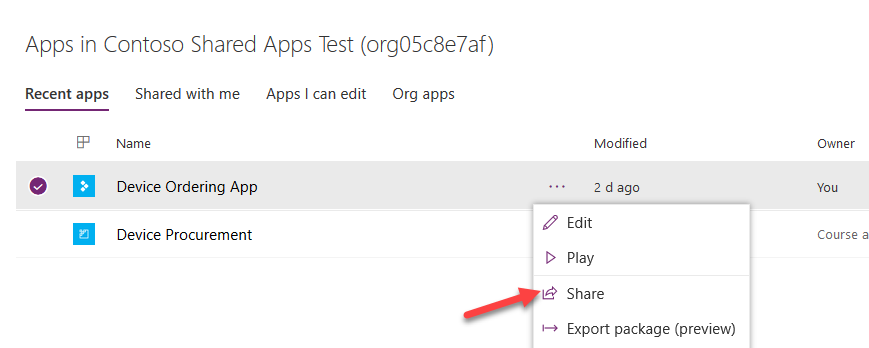
## **Exercise 6: Share the apps with the users**

**Scenario**

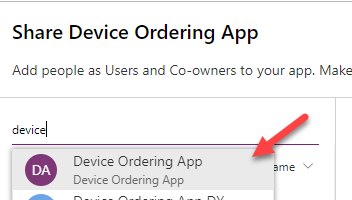
In this exercise, you will be sharing the Device Ordering canvas app with all users of the Device Ordering App Azure AD security group (we pre-created this for you!). When you do the share, you will pick the Device Ordering App security role for the CDS permissions. Behind the scenes this will cause the system to create a CDS team that is associated with the Azure AD group and associate that team with the security role. The end result of this is making it possible to give users access to the app and CDS data all in the same sharing action.

**Task 1: Share the device ordering app**

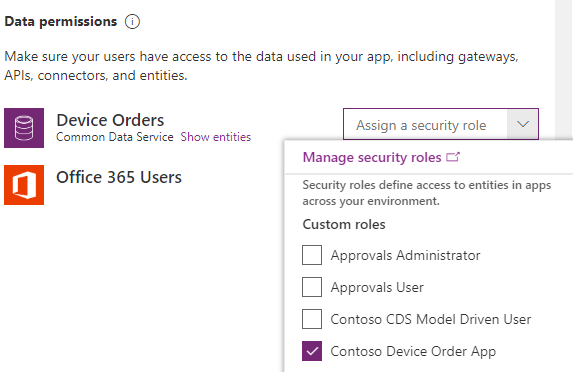
1. Navigate to https://make.powerapps.com and make sure you are in the correct environment (Central Apps Test).
2. Select Apps.
3. Click on the … More Commands button on the Device Ordering App and select Share.



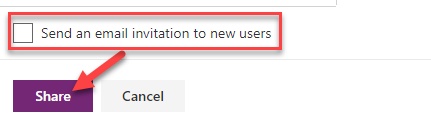
1. Search for **Device** and select the **Device Ordering App** group.



1. On the right side of the share panel next to the CDS Connector click on the Assign a security role dropdown and select the **Contoso Device Order App** security role. The system will create a CDS Team and assign this security role to the team. Any member of the Azure AD Group will then get those privileges.



1. Un-check the Send Notification checkbox and Click on the Share button.



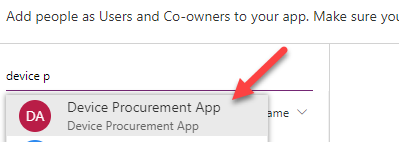
1. Don’t Close the share pane we are going to use it in the next task.

**Task 2: Share the device procurement model-driven app**

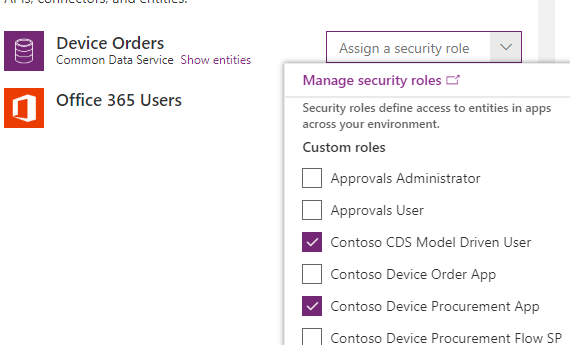
While you can click on the Share link of a model-driven app it will simply list the manual steps you need to take. These steps are the following \* Make sure the model-driven app has a security role assigned to it, and users that need to run it has that role \* Make sure the users that need to access the app have a security role giving them appropriate CDS permissions to the data.

In the future, the model-driven share experience will be updated to be more like the canvas experience above. Until that happens, we can leverage the canvas share to do the connection of an Azure AD group to a CDS Team. In our example, we want the back office group to also have access to our canvas app, but if we didn’t after using the share to hookup the team and the security role to the Azure AD group we could remove the share to this group and it would leave the team and connection to Azure AD in place.

1. While still on the share pane from the prior task, search for the **Device Procurement App** group and select it.



1. In the Assign security role drop down select the **Contoso CDS Model Driven User** and **Contoso Device Procurement App** security roles.



1. Un-check send an email and click **Share**. Since our model-driven app uses the Contoso Device Procurement App role, our back office users will now have access to the model-driven app.
2. Close the share pane.

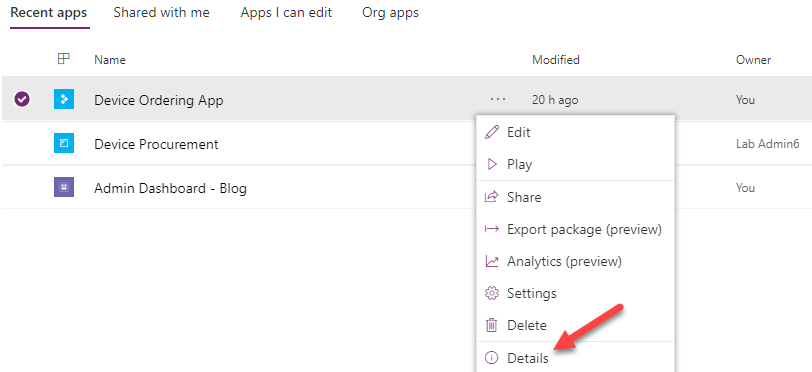
## **Exercise 7 (Optional): Test the application**

**Scenario**

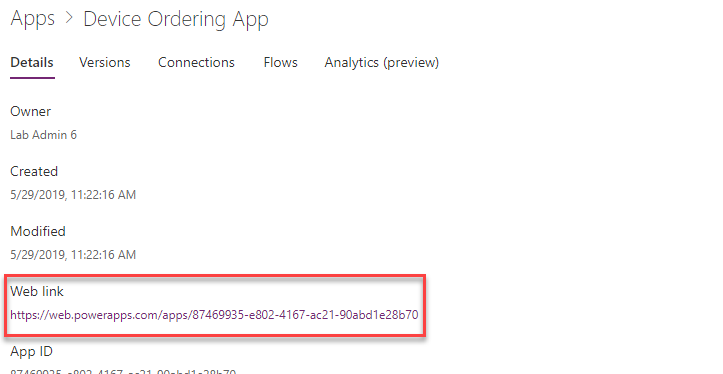
In this exercise, you will login with the users and test the application.

**Task 1: Test the Device Ordering app**

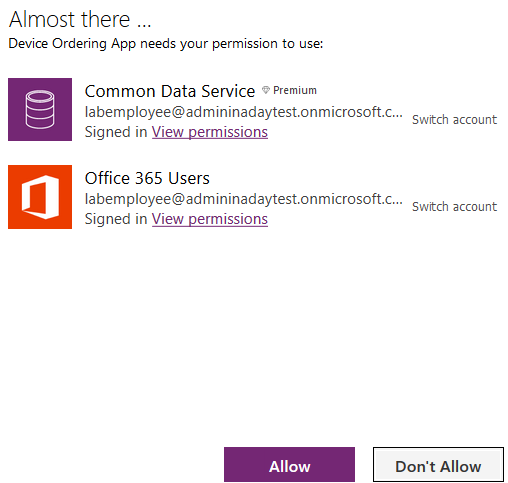
1. Navigate to make.powerapps.com and make sure you are in the Central Apps Test environment.
2. Select on **Apps**.
3. Click on … next to the **Device Ordering App** and select **Details**.



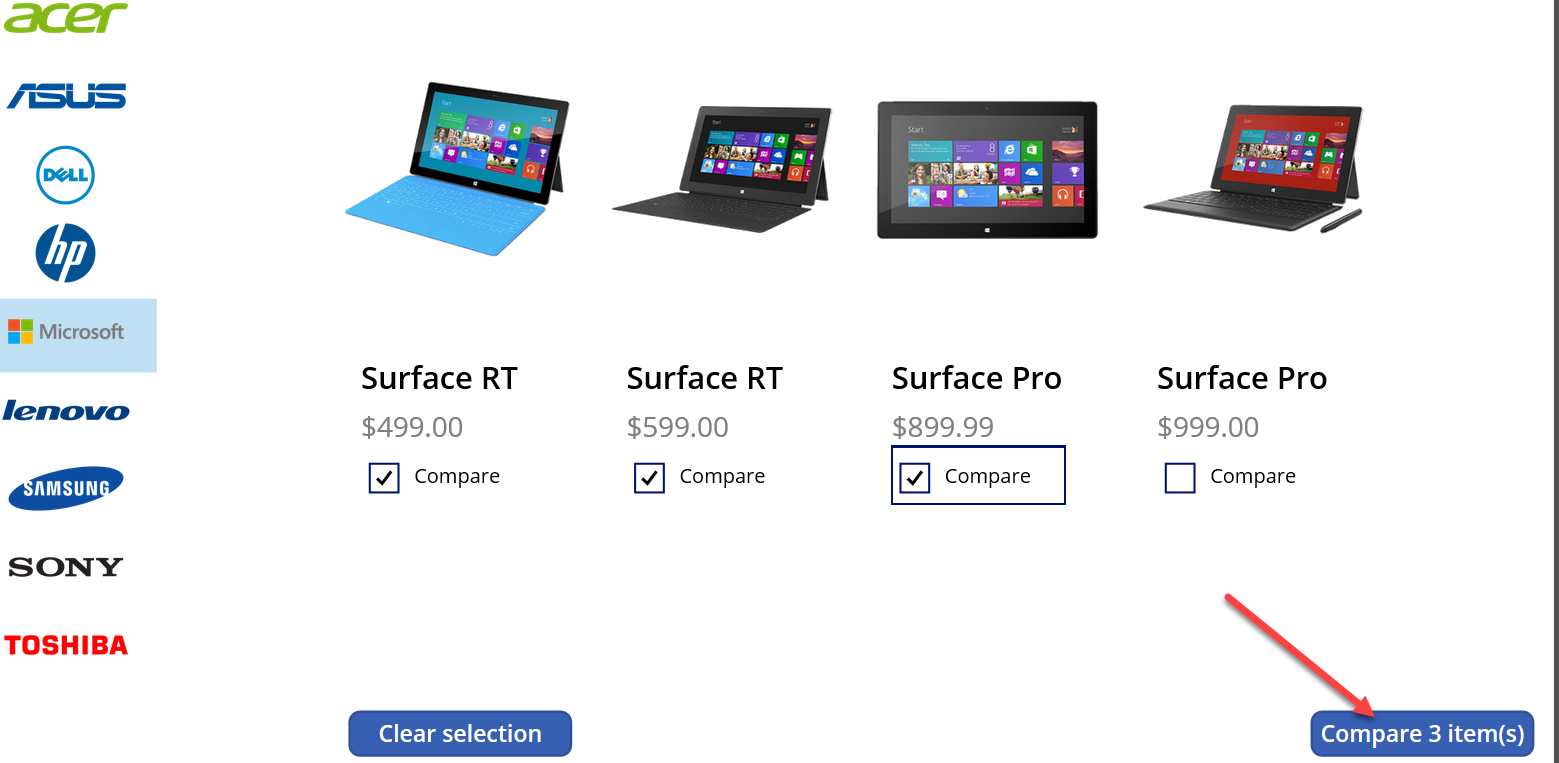
1. On the details page copy the URL in the Web link section.



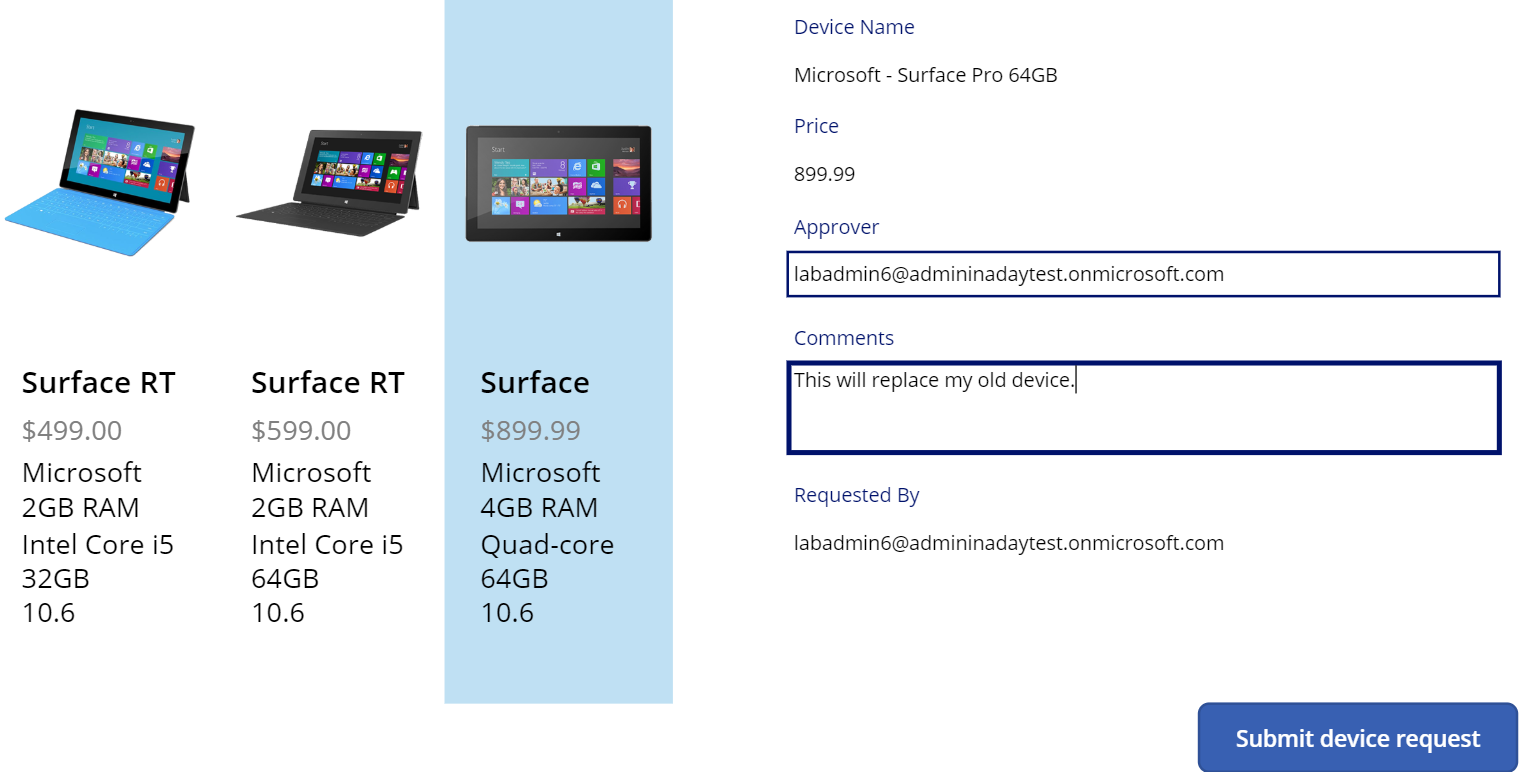
1. In another in-private browser window, paste the copied URL into the browsers address bar to launch the app.
2. Log in with your Lab Employee account.
3. Make sure the selected connections are correct and click Allow.



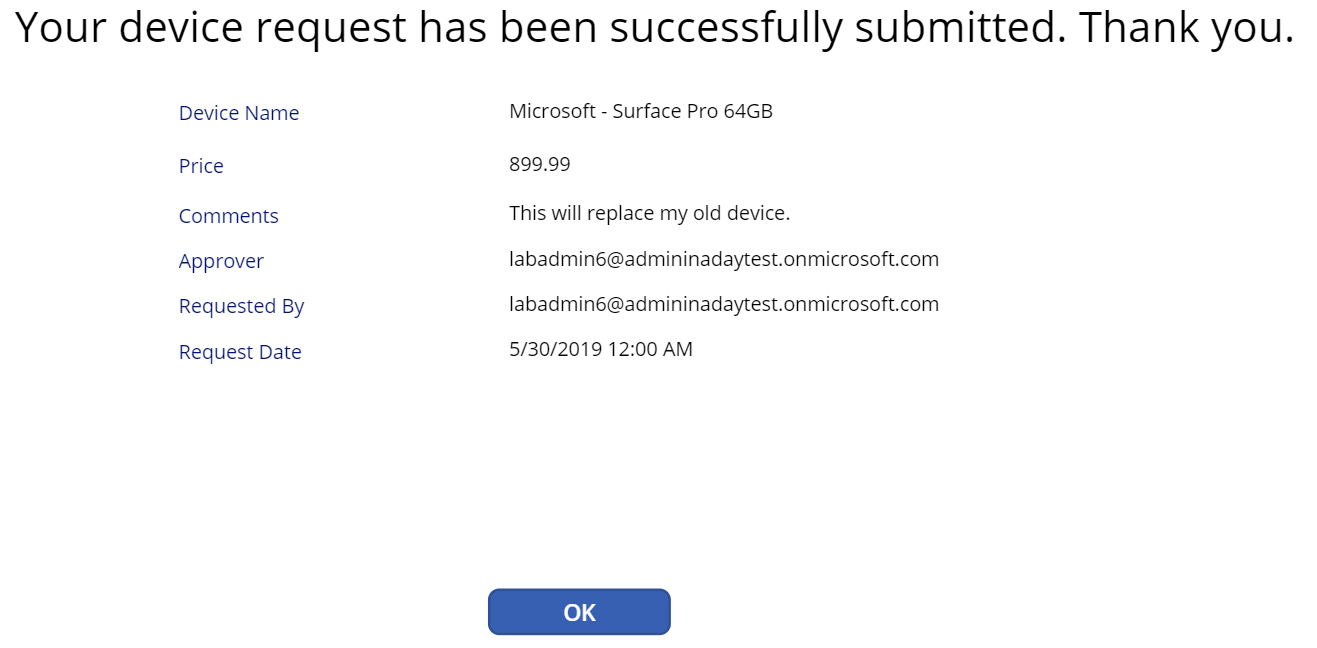
1. Select Manufacturer.
2. Select Few devices and click Compare.



1. Select one of the devices.
2. Change the approver to your lab admin account.
3. Provide a comment and click **Submit Device Request**.

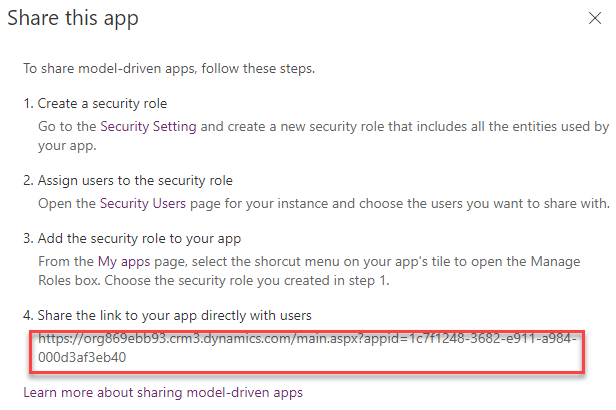


1. Click OK.

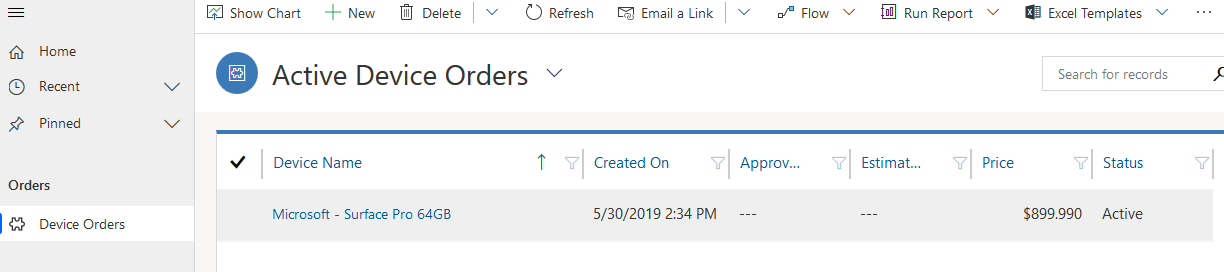


**Task 2: Test the Device Procurement app**

1. In the browser window logged in with Lab Admin, navigate to make.powerapps.com and make sure you are in the correct environment.
2. Select Apps and click on the … next to the **Device Procurement App** and select Share.
3. Copy the URL from Step 4.



1. In a private browser window paste the url for the app into the address bar and login with your Lab Back Office account. If the you can’t login, give few minutes and try again.
2. Select Device Orders and you should see at least one device order.



## **Exercise 8 (Optional): Setup the flow to use a service principal**

**Scenario**

In this exercise, you will be setting up the Power Automate flow to use a service principal and shared service accounts.

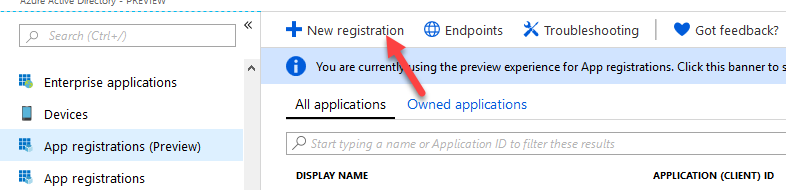
The goal of this setup is to ensure the flow runs in a non-system admin account with least privileges required.

To accomplish that the following accounts will be used \* For Common Data Service trigger and actions we will use a service principal that has just the permissions required for the flow actions \* For the Approval and Office 365 e-mail send we will use a Service Account that only has permission to send e-mail and has no other permissions at this time.

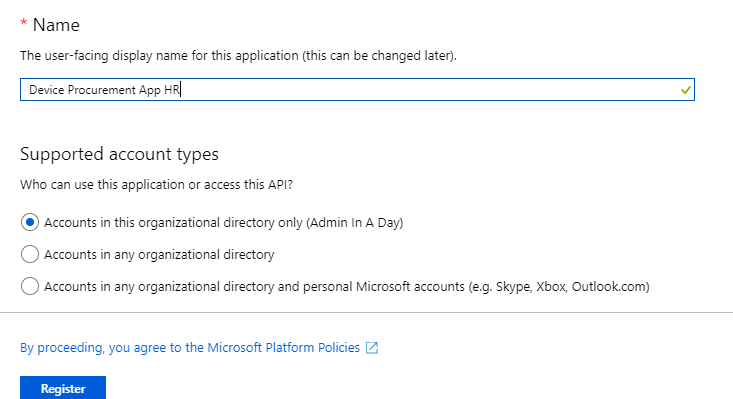
The most important thing is to make sure we don’t use our admin account that generally has very broad access to resources in the tenant.

**Task 1: Setup the service principal**

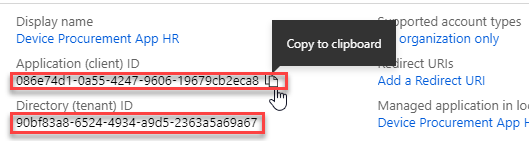
1. Navigate to https://portal.azure.com/
2. Select Azure Active Directory.
3. Select App Registration (not the one that says (Legacy)) and click New Registration.



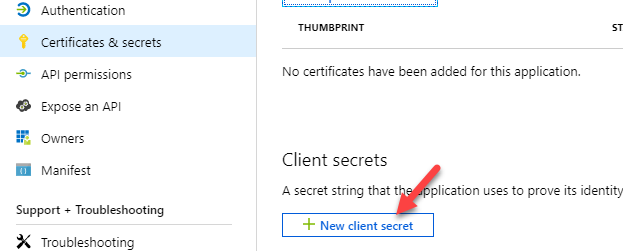
1. Enter Device Procurement App (Your Initials) for Name, select Accounts in this Organizational Directory Only, and click Register.



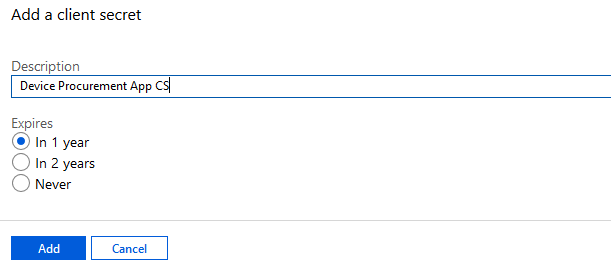
1. Copy the Application(client) ID and Directory Tenant I. Save them in a notepad.



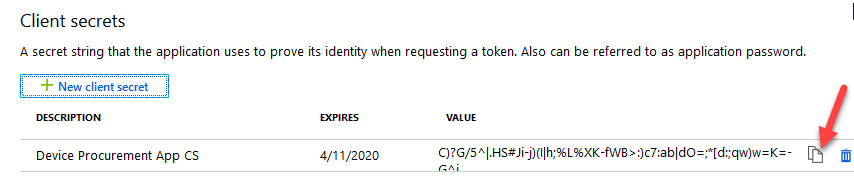
1. Select Certificates & Secrets and click + New Client Secret.



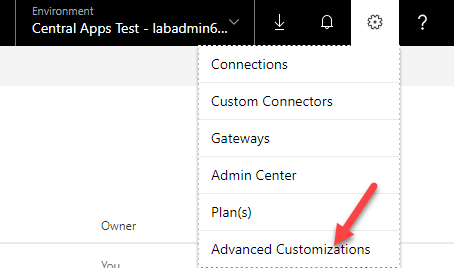
1. Enter Description and click Add. For the Expires duration, we are choosing 1 year, in a real deployment you need to make sure if you choose 1 or 2 years you have a plan to rotate the keys prior to expiration or anything using the key will fail upon expiration.



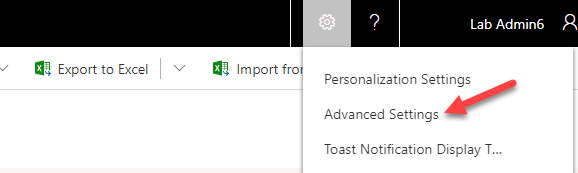
1. Copy the Value and keep in a notepad. This value is not available again once you leave this area of the portal.



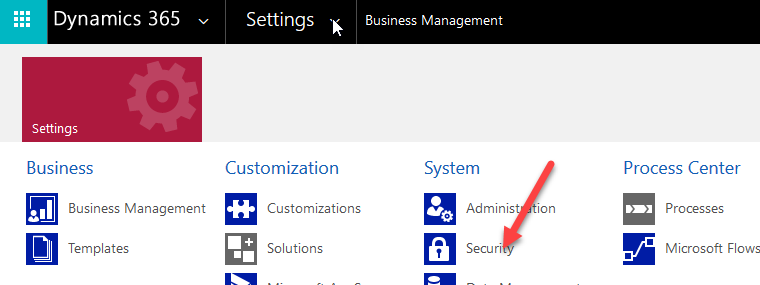
1. Go back to <https://make.powerapps.com> click Settings and select Advanced Customizations.



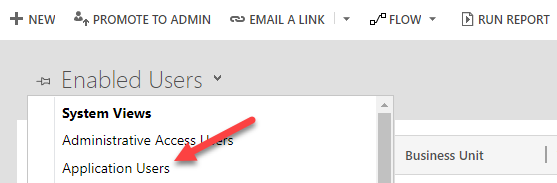
1. Select Users.
2. Select Advanced Settings.



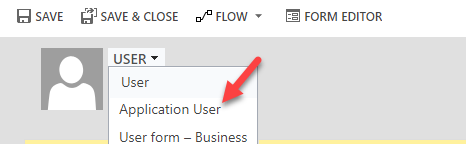
1. Navigate to Settings | Security.



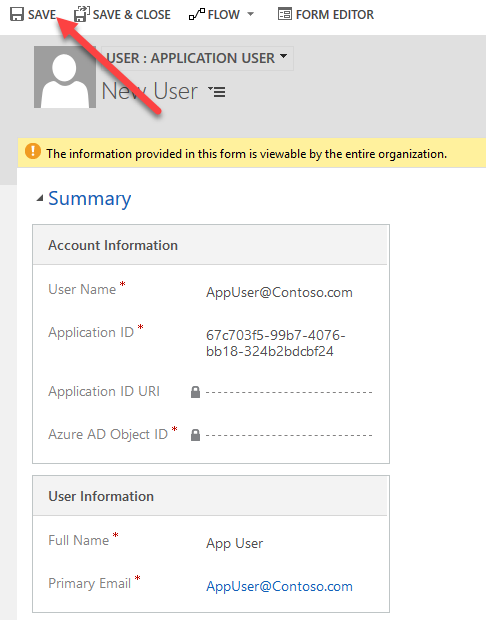
1. Click Users.
2. Change the view to Application Users.



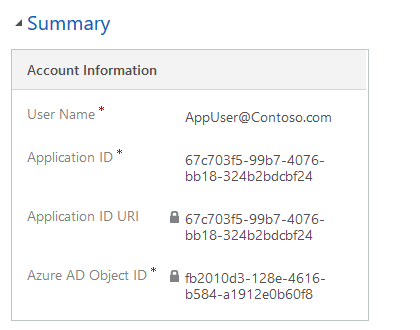
1. Click New.
2. Change the form to Application User.



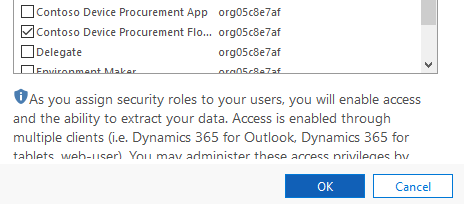
1. Enter Contoso App User for Name.
2. Paste the Application ID you copied from the Azure Portal.
3. In the Full Name field, provide First and Last Name and then click Done.
4. Provide your lab admin email address and click Save.



1. The Application ID URL and Azure AD Object ID should populate.

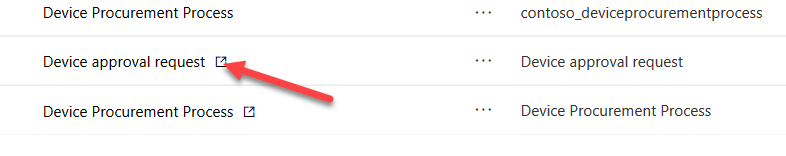


1. Click Manage Roles
2. Select the Contoso Device Procurement Flow SP security role and click OK.

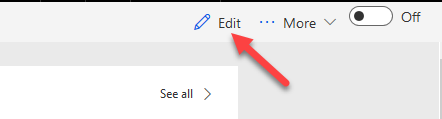


**Task 2: Edit the flow to fix up the connections**

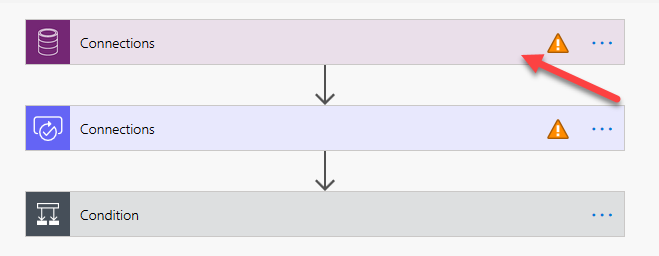
1. Navigate to <https://make.powerapps.com>
2. Select Solutions and open Contoso Device Order Management.
3. Locate Device Approval Request and click to open.



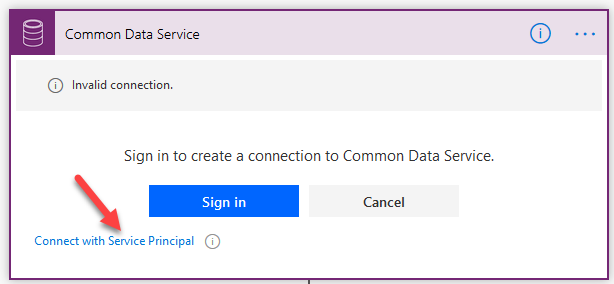
1. Click Edit.



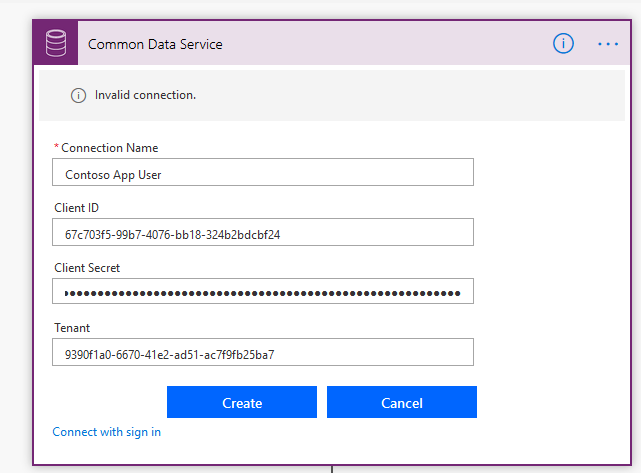
1. Click on the trigger step to expand.



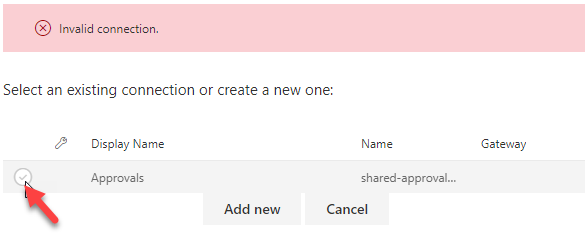
1. Click Add New.
2. Click Connect with Service Principal.



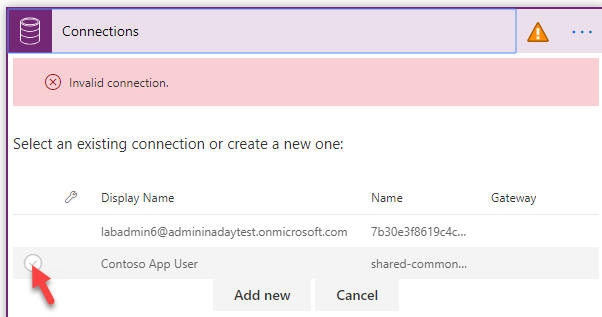
1. Provide a name Contoso App User.
2. Paste the Application (Client) ID.
3. Paste the secret from the notepad in the Client Secret field.
4. Paste the Tenant ID from the notepad in the Tenant field.
5. Click Create.



1. Click on the Approval step.
2. Select Approvals.



1. Expand the Condition stage.
2. Select Contoso App User.



1. Expand the Email action.
2. Click Add New Connection.
3. Click on the … button and select Add New Connection. This will show a login dialog.
4. Login with the Service Account credentials which are [serviceaccount@yourtenant.onmicrosoft.com](mailto:serviceaccount@yourtenant.onmicrosoft.com) – replace the @yourtenant.onmicrosoft.com with your lab admin tenant name. The password will be the same as your lab admin account.
5. In the If No branch, click on the CDS connector and select the Contoso App User service principal.
6. In the If No branch, click on the Email action and select the Service Account connection.
7. Click on the Flow Checker button.
8. Make sure you don’t have errors and close the Flow Checker pane.
9. Save the Flow.
10. Click on the Back arrow next to the Flow Name.
11. Confirm the flow is now turned on (upper right corner) and turn on the Flow if it is still turned off.

