PowerApps

Power Platform Build Tools Hands-on Lab

Module 2

Automating Solution Deployment Using Azure DevOps

Last Updated: July 10, 2019

Authors: Per Mikkelsen, Shan McArthur, Evan Chaki

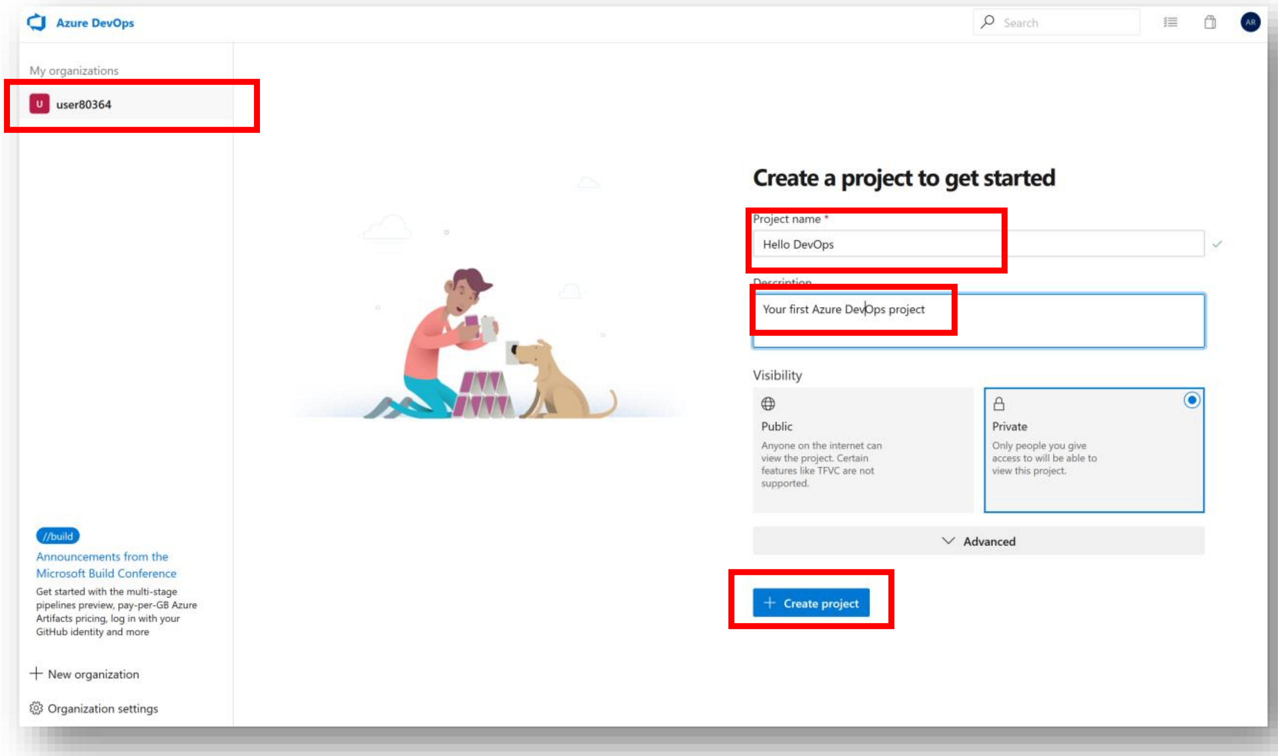
Lab Scenario

In this hands-on lab, you will create an Azure DevOps project, setup permissions and create the required pipelines to automatically export your app (as an unmanaged solution) from a development environment, generate a build artifact (managed solution) and finally deploy the app into production. The lab will also introduce you to Microsoft Power Platform Build Tools.

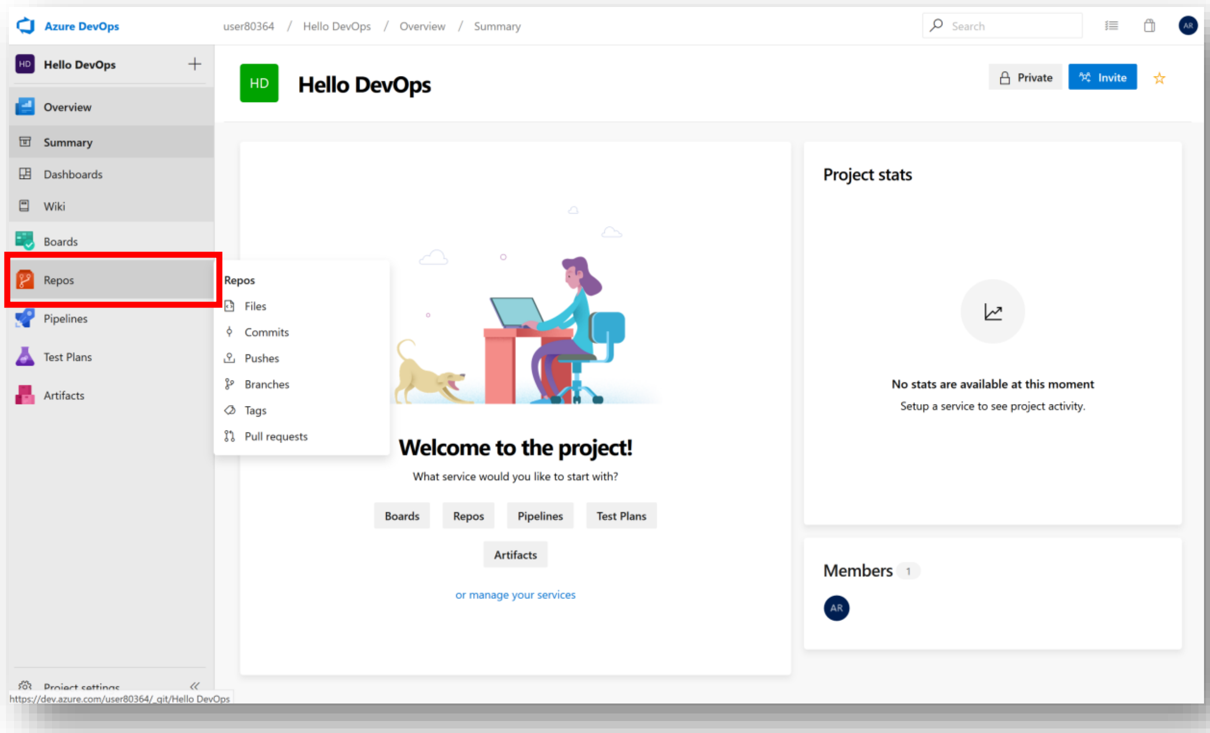
Create an Azure DevOps Project

We are going to use Azure DevOps for both source code storage and build and deployment automation. You can use any automated source control and build automation tools using the same principles. Your configuration data should be exported from a development environment, processed by Package Deployer, and checked into source control.

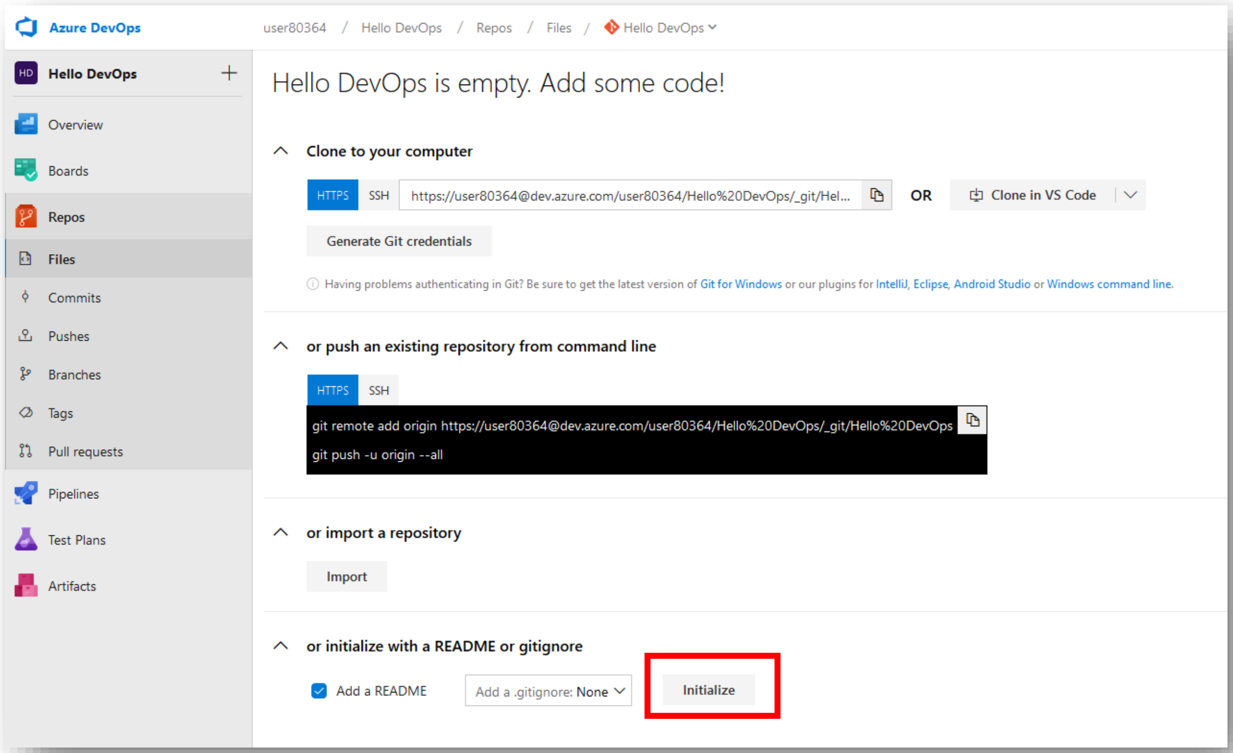
1. Log into dev.azure.com with your credentials and click on your organization in the left panel. Follow the instructions to create a project. Click Create Project.


Azure DevOps
Search
My organizations
user80364
Create a project to get started
Project name *
Hello DevOps
Description
Your first Azure DevOps project
Visibility
Public
Anyone on the internet can view the project. Certain features like TFVC are not supported.
Private
Only people you give access to will be able to view this project.
Advanced
Create project
//build
Announcements from the Microsoft Build Conference
Get started with the multi-stage pipelines preview, pay-per-GB Azure Artifacts pricing, log in with your GitHub identity and more
New organization
Organization settings

2. When the project is completed, you will need to create a Repo to hold the source code. Click on the Repos link in the left navigation.



3. Initialize the repo with the default README by clicking the Initialize button.



Hello DevOps is empty. Add some code!

Clone to your computer

HTTPS SSH https://user80364@dev.azure.com/user80364/Hello%20DevOps/_git/Hel... OR Clone in VS Code

Generate Git credentials

Having problems authenticating in Git? Be sure to get the latest version of Git for Windows or our plugins for IntelliJ, Eclipse, Android Studio or Windows command line.

or push an existing repository from command line

HTTPS SSH

```
git remote add origin https://user80364@dev.azure.com/user80364/Hello%20DevOps/_git/Hello%20DevOps
git push -u origin --all
```

or import a repository

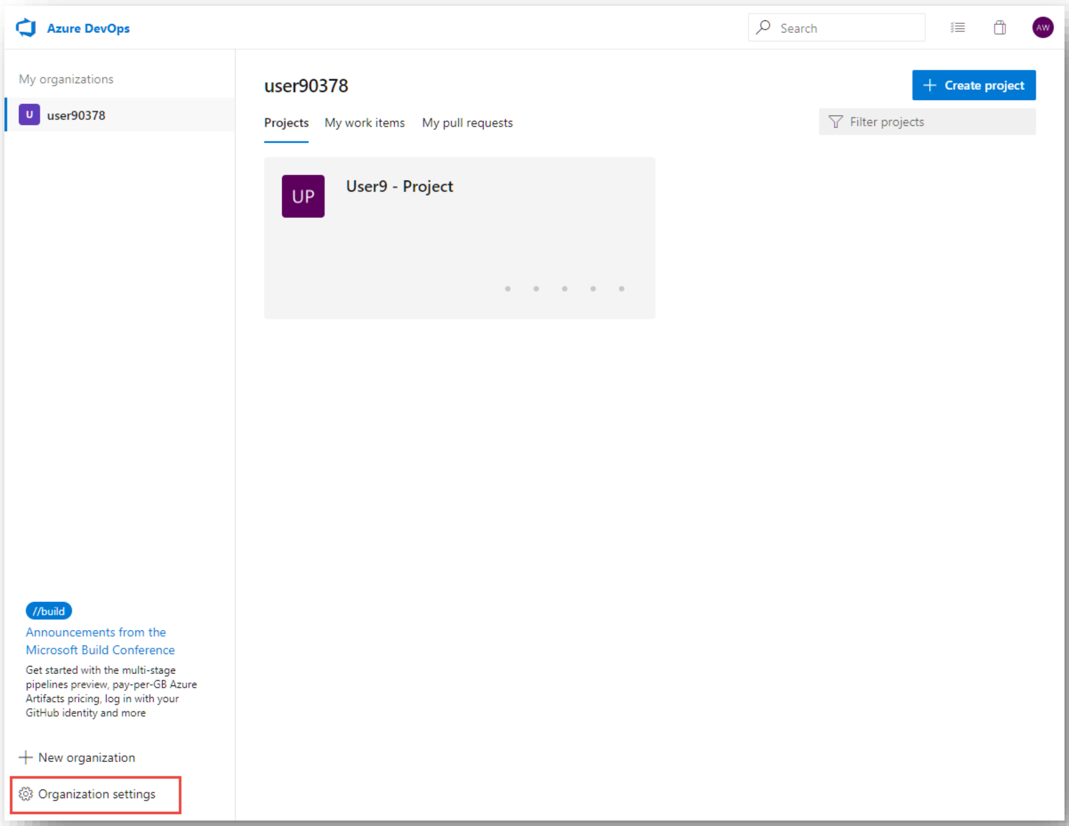
Import

or initialize with a README or gitignore

☑ Add a README Add a .gitignore: None Initialize

Install and Enable the Azure DevOps Extensions for Power Platform

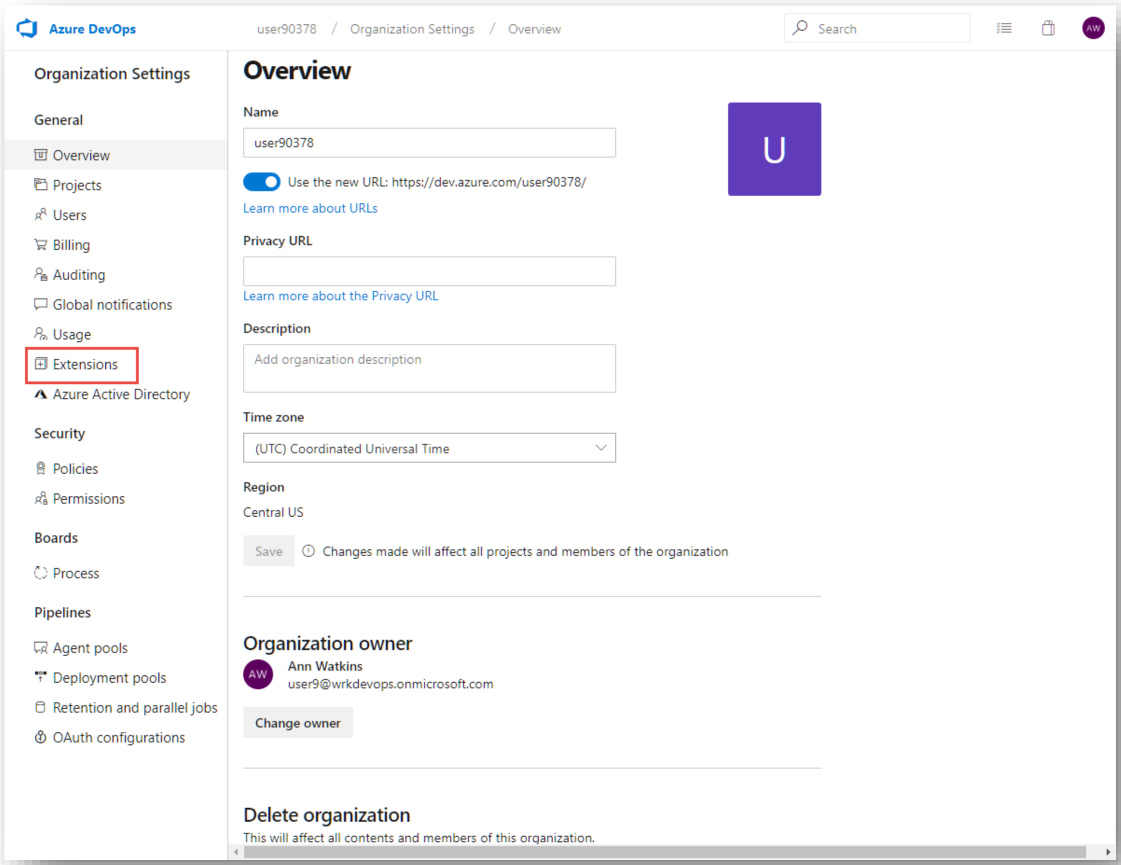
4. On the organization page, click the Organization Settings link in the bottom left of the navigation panel.



Announcements from the Microsoft Build Conference

Get started with the multi-stage pipelines preview, pay-per-GB Azure Artifacts pricing, log in with your GitHub identity and more

5. On the organization settings page, click the Extensions link in the left navigation panel.



Azure DevOps
user90378 / Organization Settings / Overview
Search

Organization Settings

General
- Overview
- Projects
- Users
- Billing
- Auditing
- Global notifications
- Usage
- Extensions
- Azure Active Directory

Security
- Policies
- Permissions

Boards
- Process

Pipelines
- Agent pools
- Deployment pools
- Retention and parallel jobs
- OAuth configurations

Overview

Name: user90378

Use the new URL: https://dev.azure.com/user90378/

Learn more about URLs

Privacy URL

Learn more about the Privacy URL

Description: Add organization description

Time zone: (UTC) Coordinated Universal Time

Region: Central US

Save — Changes made will affect all projects and members of the organization

Organization owner

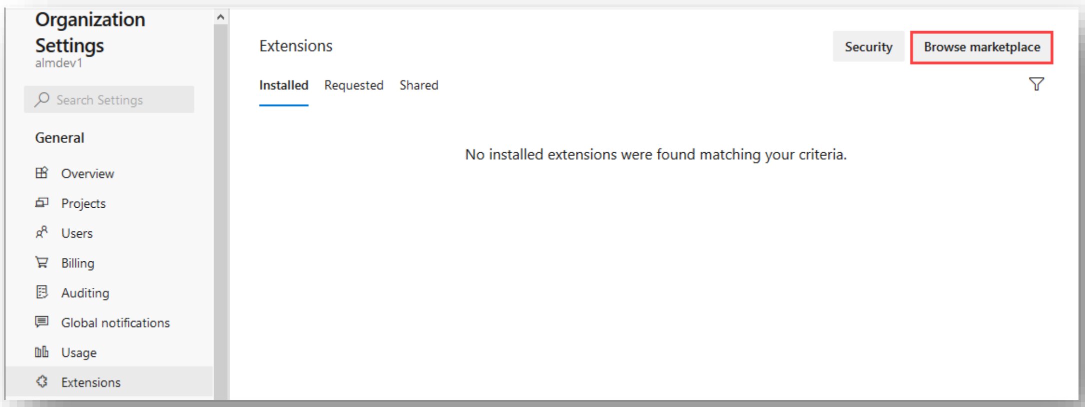
Ann Watkins
user9@wrkdevops.onmicrosoft.com

Change owner

Delete organization

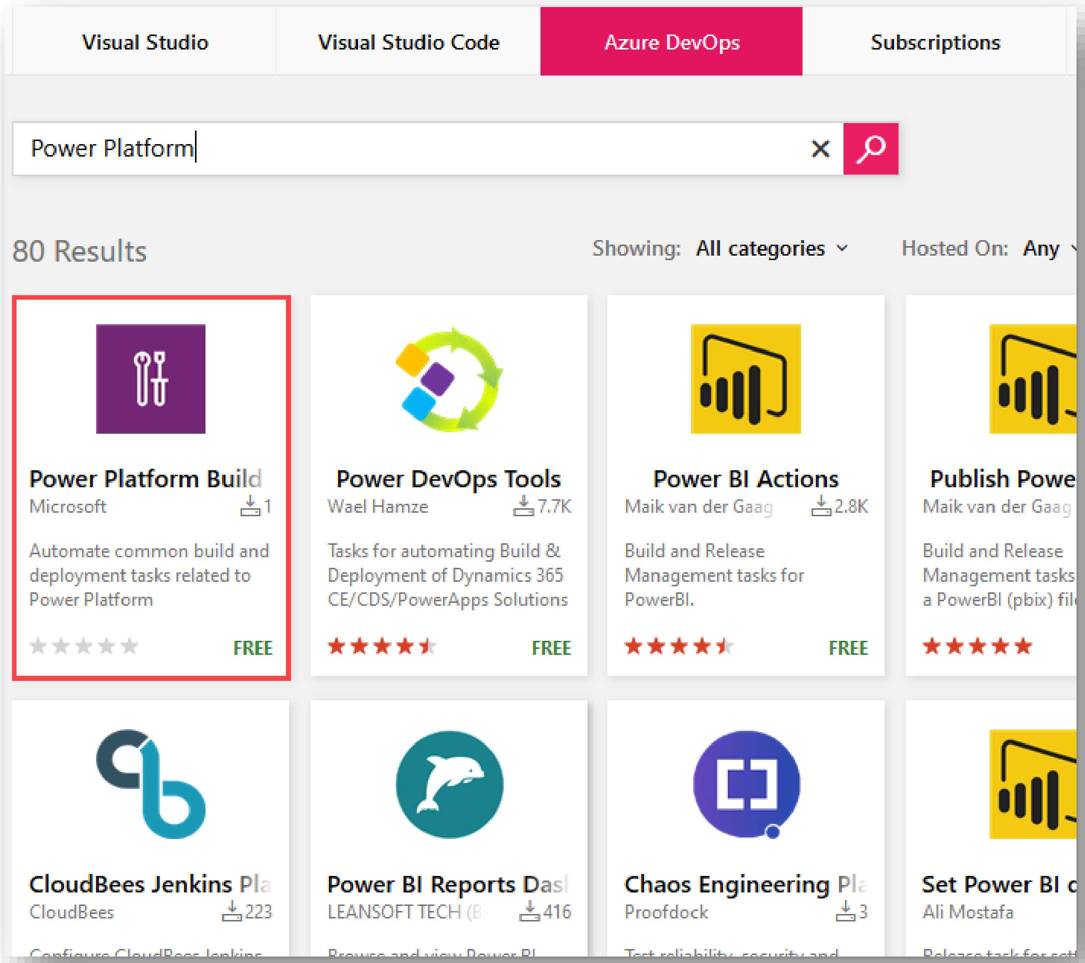
This will affect all contents and members of this organization.

6. Click ‘Browse marketplace’.



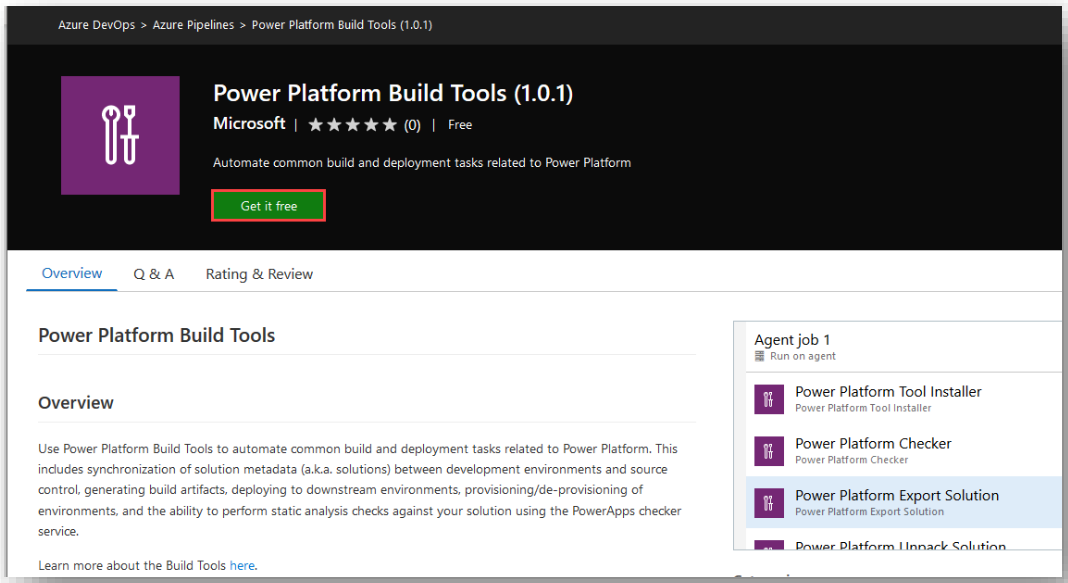
No installed extensions were found matching your criteria.

7. This will redirect you to the Azure DevOps marketplace. Search for Power Platform and select the Power Platform Build Tools. Note that there are great community options too.



Power Platform Build	Power DevOps Tools	Power BI Actions	Publish Powe
Microsoft 1	Wael Hamze 7.7K	Maik van der Gaag 2.8K	Maik van der Gaag
Automate common build and deployment tasks related to Power Platform	Tasks for automating Build & Deployment of Dynamics 365 CE/CDS/PowerApps Solutions	Build and Release Management tasks for PowerBI.	Build and Release Management tasks a PowerBI (pbix) fil
☆☆☆☆☆ FREE	★★★★½ FREE	★★★★½ FREE	★★★★★
CloudBees Jenkins Pla	Power BI Reports Das	Chaos Engineering Pla	Set Power BI c
CloudBees 223	LEANSOFT TECH (B 416	Proofdock 3	Ali Mostafa

8. Click ‘Get it free’.



Azure DevOps > Azure Pipelines > Power Platform Build Tools (1.0.1)

Power Platform Build Tools (1.0.1)

Microsoft | ★★★★★ (0) | Free

Automate common build and deployment tasks related to Power Platform

Power Platform Build Tools

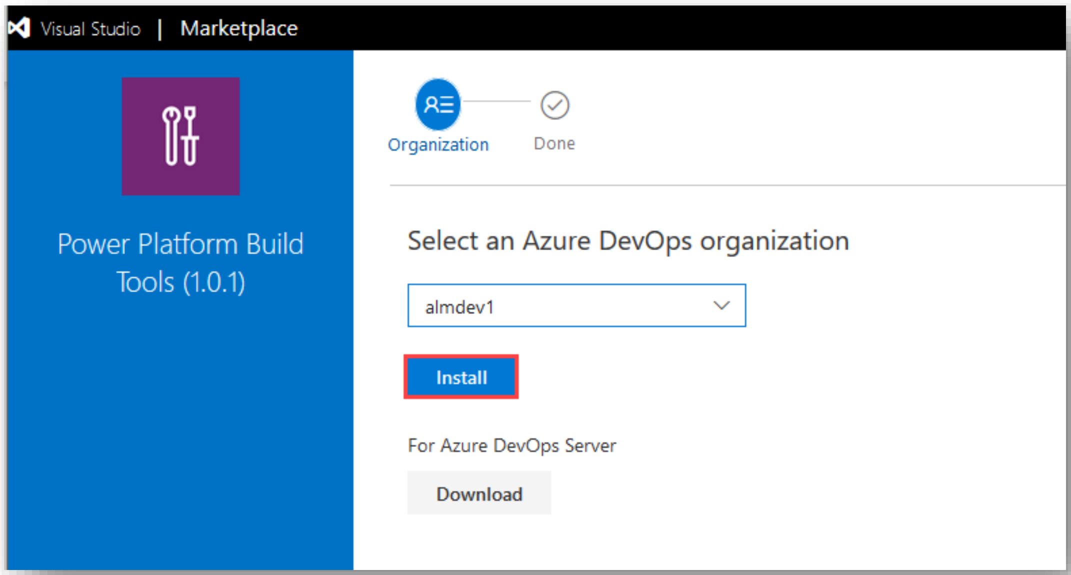
Overview

Use Power Platform Build Tools to automate common build and deployment tasks related to Power Platform. This includes synchronization of solution metadata (a.k.a. solutions) between development environments and source control, generating build artifacts, deploying to downstream environments, provisioning/de-provisioning of environments, and the ability to perform static analysis checks against your solution using the PowerApps checker service.

Learn more about the Build Tools here.

Agent job 1
Run on agent
Power Platform Tool Installer
Power Platform Tool Installer
Power Platform Checker
Power Platform Checker
Power Platform Export Solution
Power Platform Export Solution
Power Platform Unpack Solution

9. Click ‘Install’.



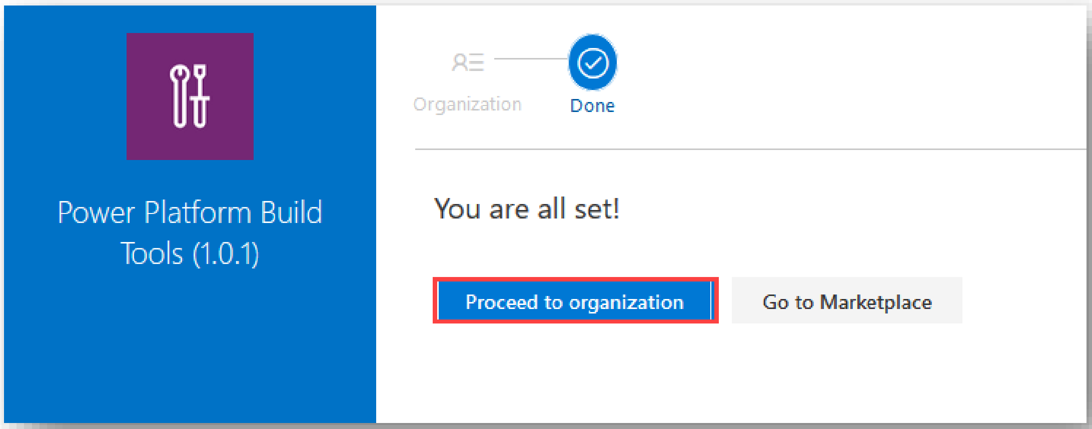
Visual Studio | Marketplace
Power Platform Build Tools (1.0.1)
Organization
Done

Select an Azure DevOps organization

almdev1

For Azure DevOps Server

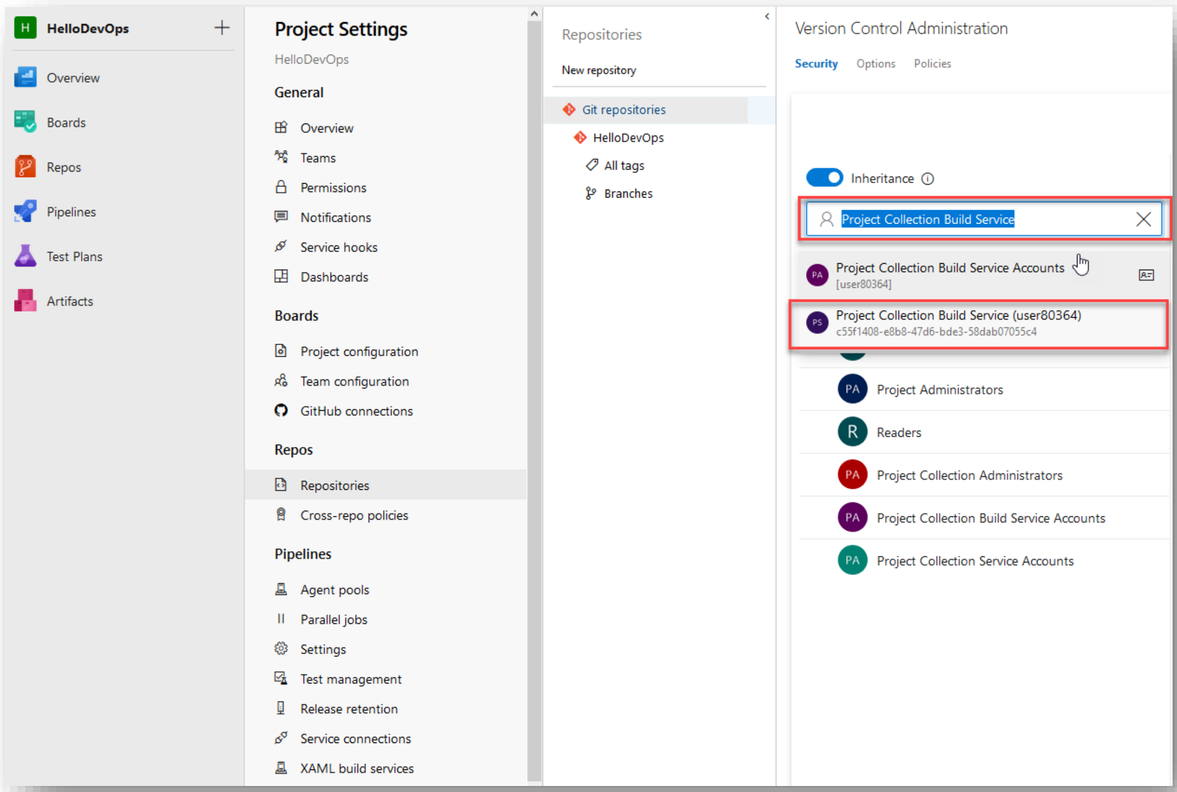
10. Click ‘Proceed to organization’.



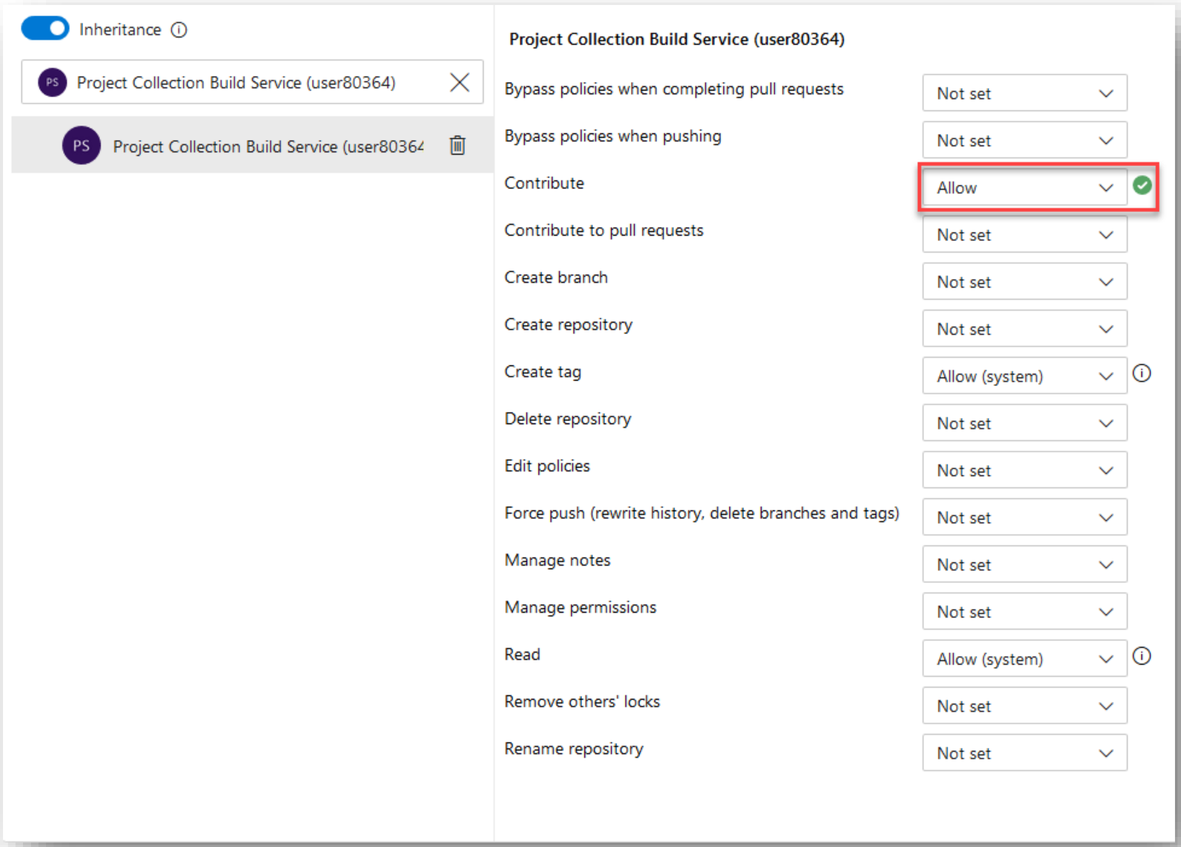
Configure Azure DevOps Permissions for Build Service Account

The build pipelines that we set up later will be exporting files from an org and checking them into your source code repo. This is not a default permission of Azure DevOps, so we need to configure the appropriate permissions for the pipeline to function properly.

11. Click the Project Settings icon on the lower left of the screen and click the Repositories link in the fly out menu. Type in **Project Collection Build Service** in the search box and select it (select the project collection with the username appended).



12. The setting for the user is displayed. Select **Allow** for 'Contribute' and ensure that the green checkbox is displayed.



Inheritance

Project Collection Build Service (user80364)

Project Collection Build Service (user80364

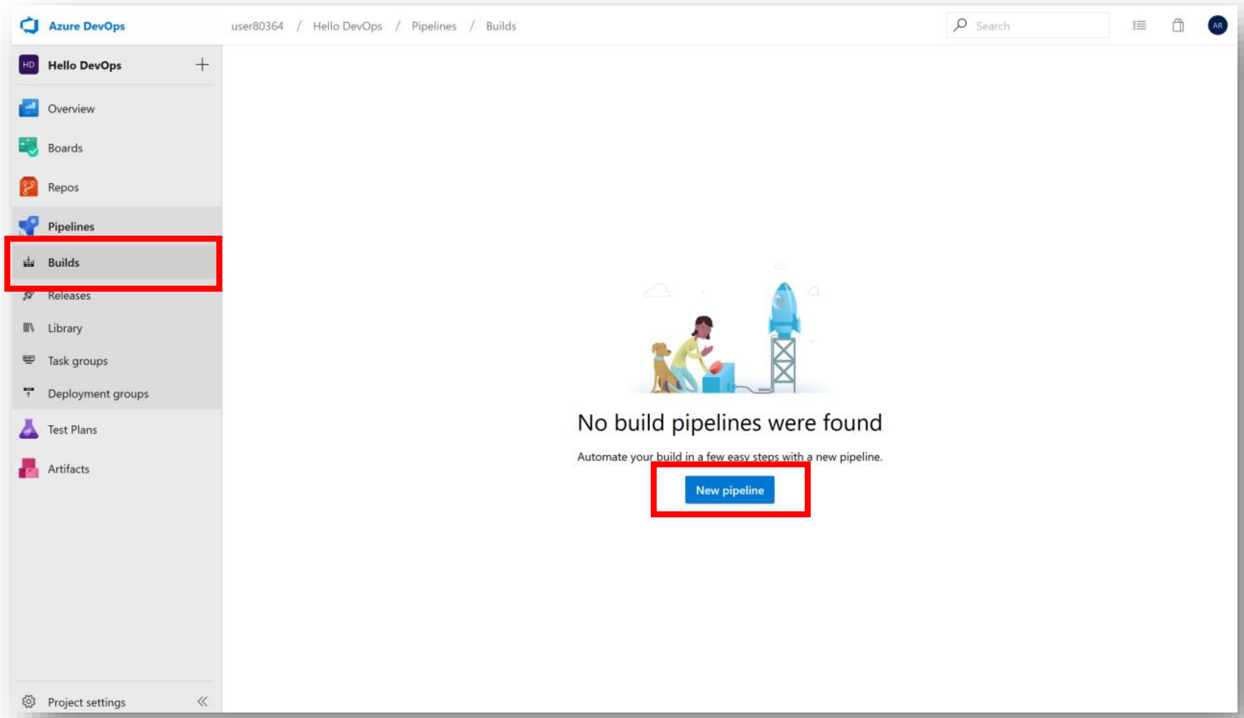
Project Collection Build Service (user80364)

Bypass policies when completing pull requests	Not set
Bypass policies when pushing	Not set
Contribute	Allow ✓
Contribute to pull requests	Not set
Create branch	Not set
Create repository	Not set
Create tag	Allow (system)
Delete repository	Not set
Edit policies	Not set
Force push (rewrite history, delete branches and tags)	Not set
Manage notes	Not set
Manage permissions	Not set
Read	Allow (system)
Remove others' locks	Not set
Rename repository	Not set

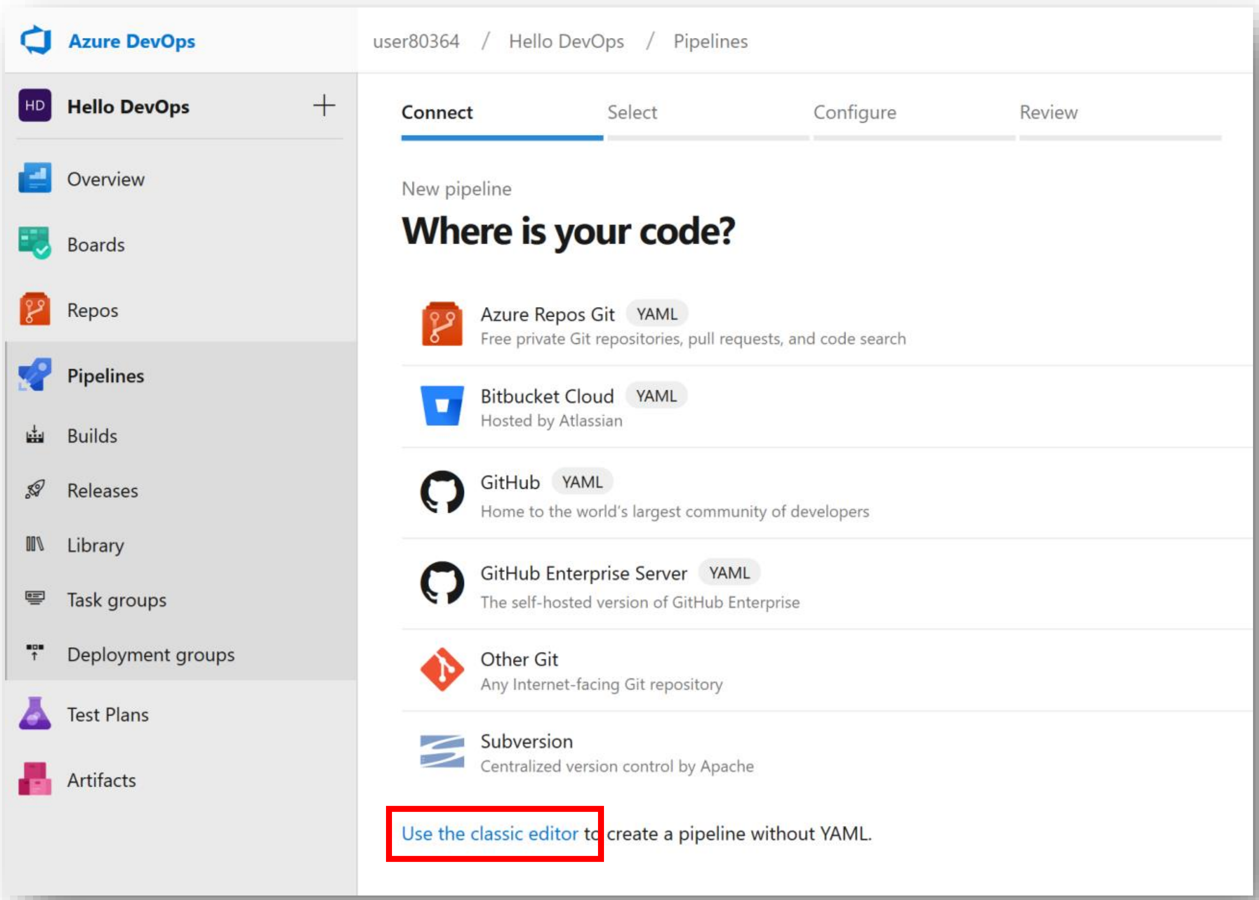
Build Pipeline 1: Create Export from Dev

The first pipeline you will create will export your solution from your development environment as an unmanaged solution, unpack it and check it into source control (your repo).

13. Click the New Pipeline button.



14. Click Use the classic editor link to create a pipeline.



New pipeline

Where is your code?

Azure Repos Git YAML
Free private Git repositories, pull requests, and code search

Bitbucket Cloud YAML
Hosted by Atlassian

GitHub YAML
Home to the world's largest community of developers

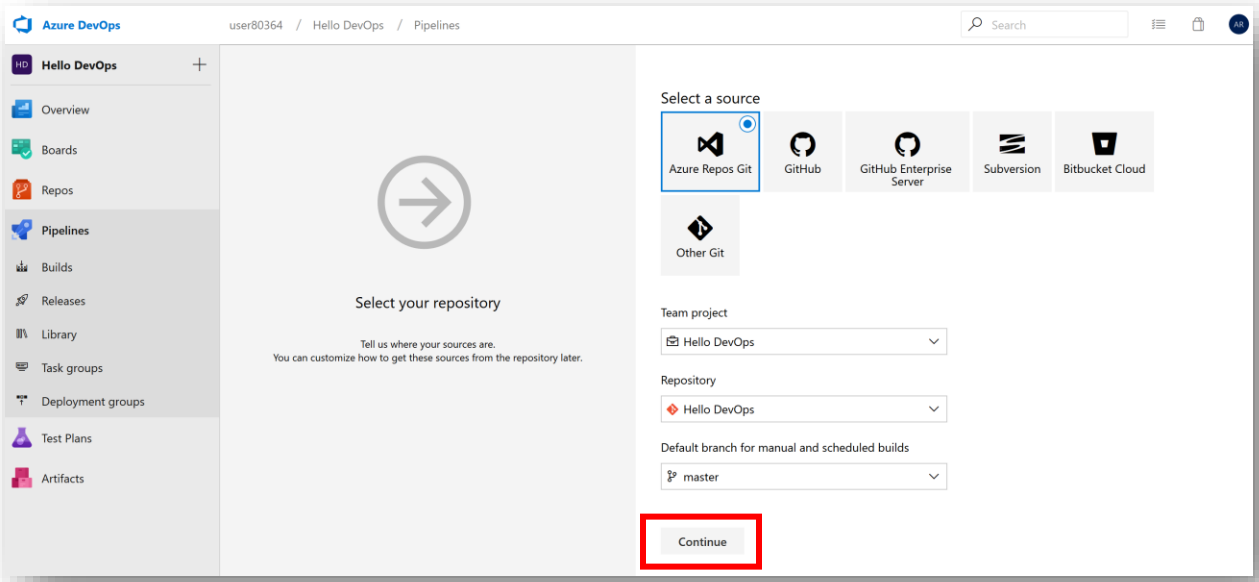
GitHub Enterprise Server YAML
The self-hosted version of GitHub Enterprise

Other Git
Any Internet-facing Git repository

Subversion
Centralized version control by Apache

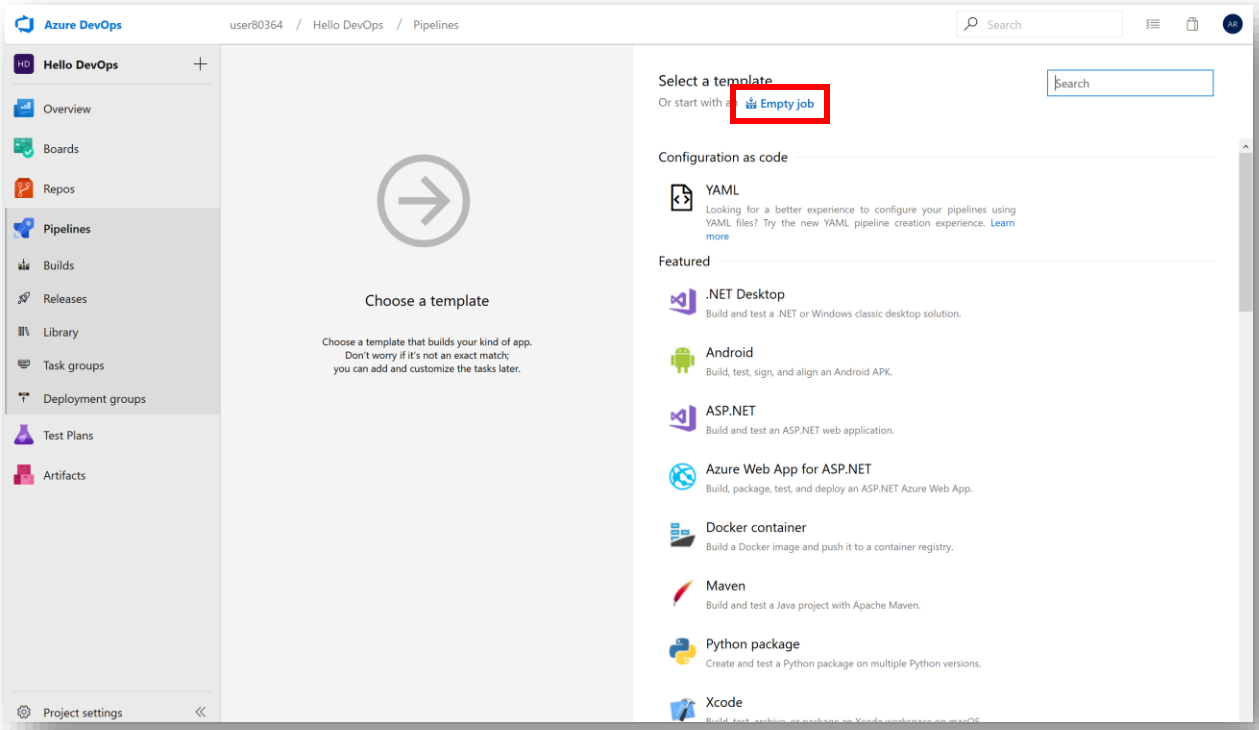
Use the classic editor to create a pipeline without YAML.

15. Leave default values as is and click ‘Continue’.



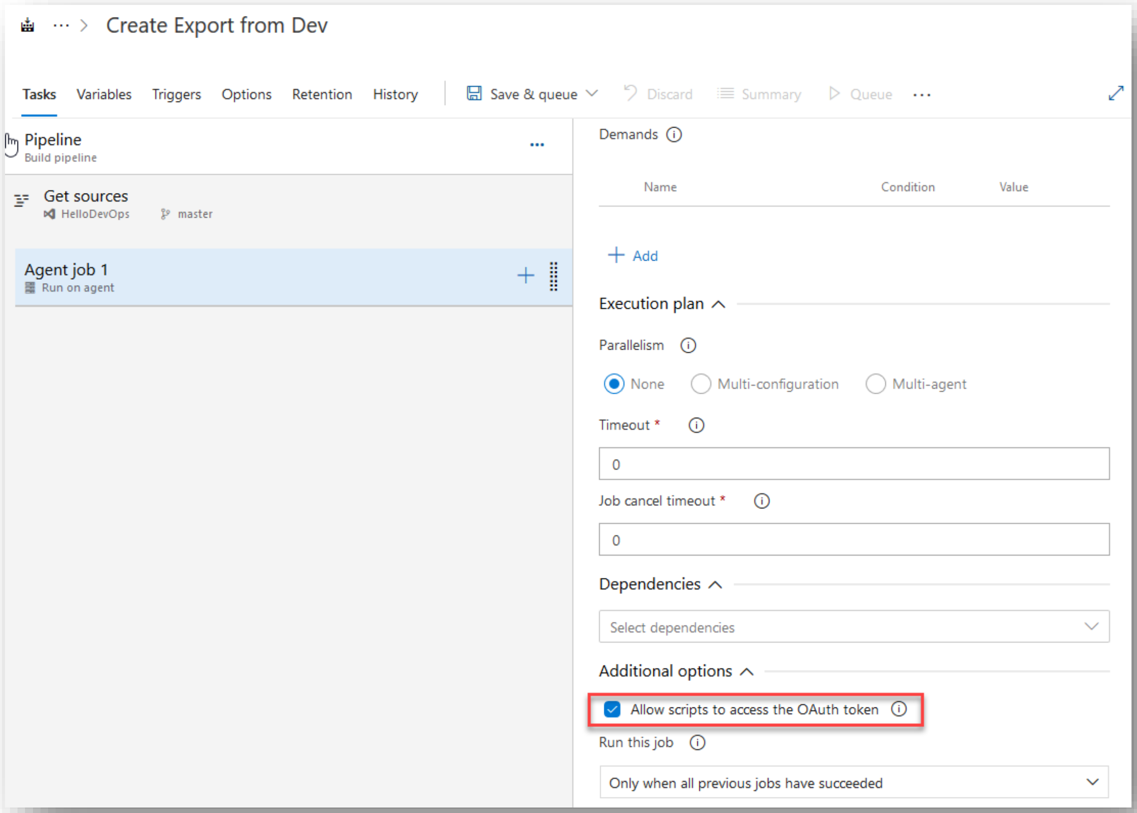
Team project: Hello DevOps
Repository: Hello DevOps
Default branch for manual and scheduled builds: master

16. Select ‘Empty Job’ to create an empty job.

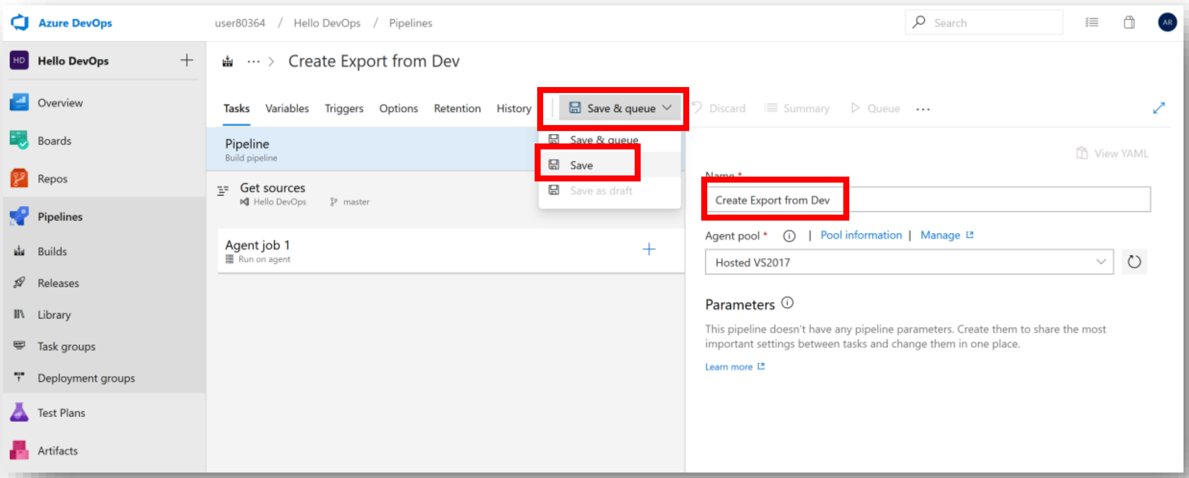


Azure DevOps
user80364 / Hello DevOps / Pipelines
Search
Hello DevOps
Overview
Boards
Repos
Pipelines
Builds
Releases
Library
Task groups
Deployment groups
Test Plans
Artifacts
Project settings
Choose a template
Choose a template that builds your kind of app.
Don't worry if it's not an exact match;
you can add and customize the tasks later.
Select a template
Or start with an Empty job
Search
Configuration as code
YAML
Looking for a better experience to configure your pipelines using YAML files? Try the new YAML pipeline creation experience. Learn more
Featured
.NET Desktop
Build and test a .NET or Windows classic desktop solution.
Android
Build, test, sign, and align an Android APK.
ASP.NET
Build and test an ASP.NET web application.
Azure Web App for ASP.NET
Build, package, test, and deploy an ASP.NET Azure Web App.
Docker container
Build a Docker image and push it to a container registry.
Maven
Build and test a Java project with Apache Maven.
Python package
Create and test a Python package on multiple Python versions.
Xcode

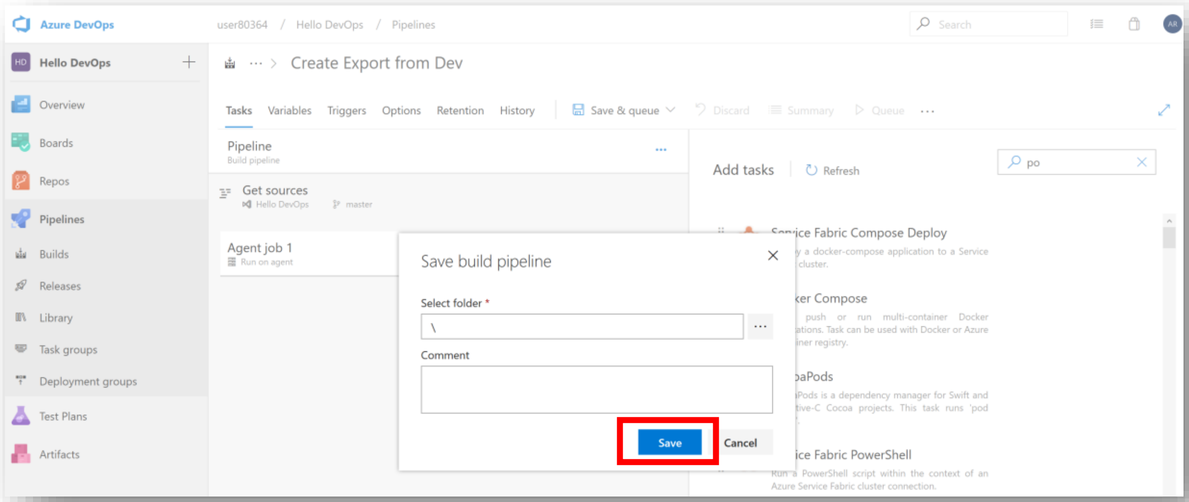
17. Select 'Agent job 1' and enable the **Allow scripts to access the OAuth token** option.



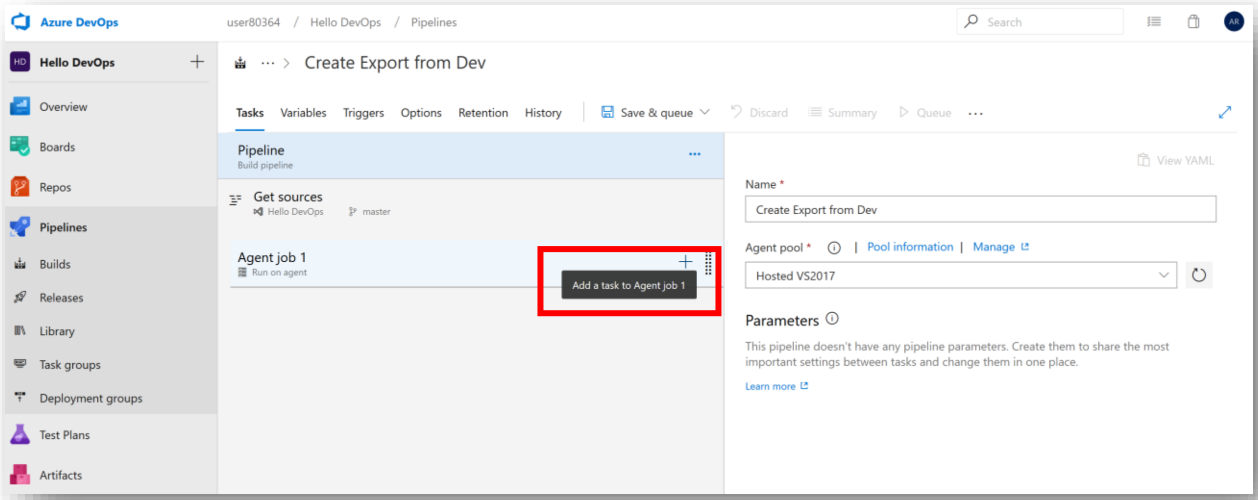
18. Name the pipeline 'Create Export from Dev' and click Save.



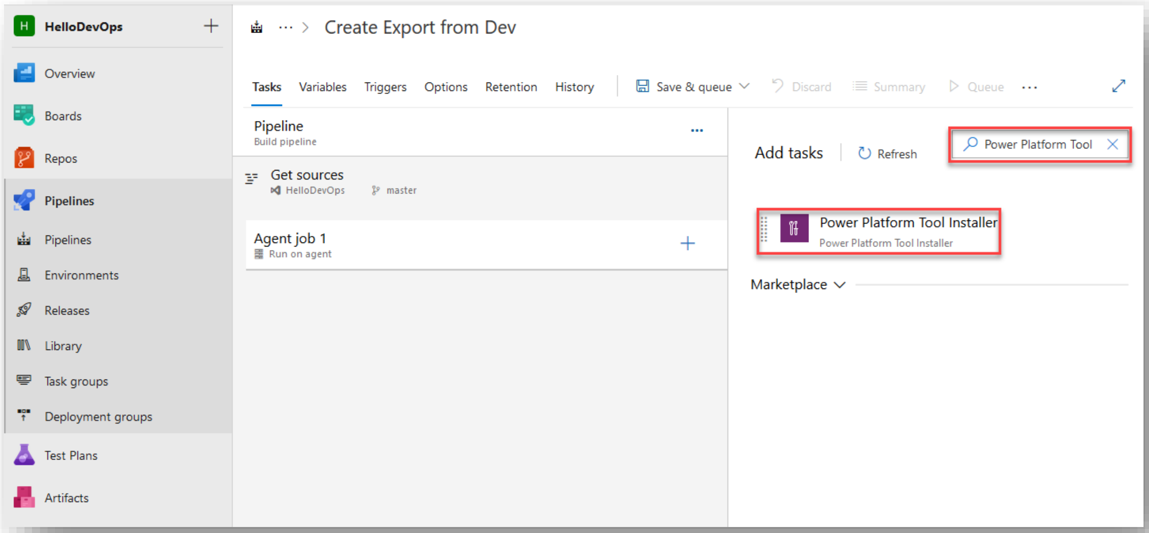
19. Changes to your pipeline are also checked into source control, so select the folder, type a comment, and click Save.



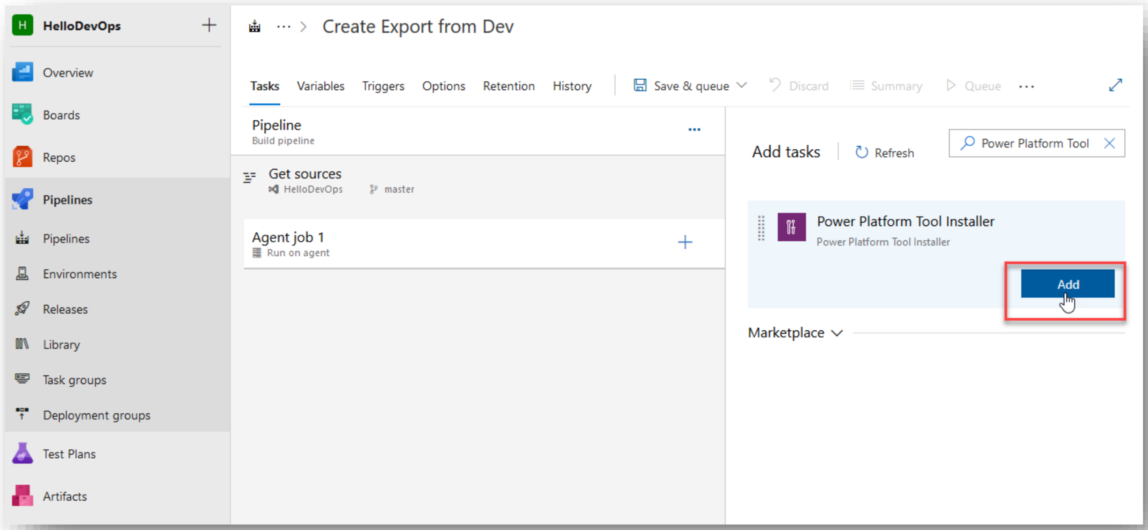
20. You are now ready to start using the Power Platform Build Tools tasks for Azure DevOps in your pipeline. Select 'Add a task to Agent job 1'.



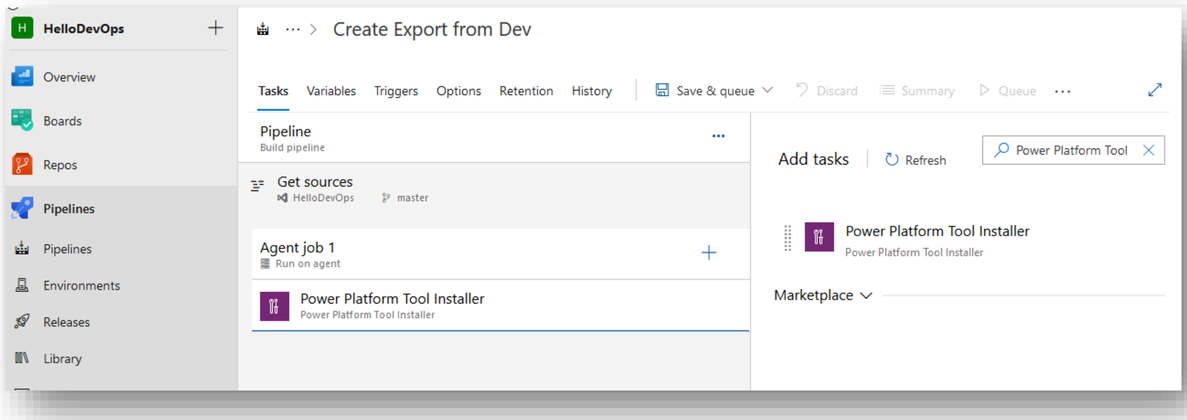
21. The very first task you will is a task that install the required tools on the agent. Search for ‘Power Platform Tool’.



22. Select ‘Add’ under the Power Platform Tool Installer task.



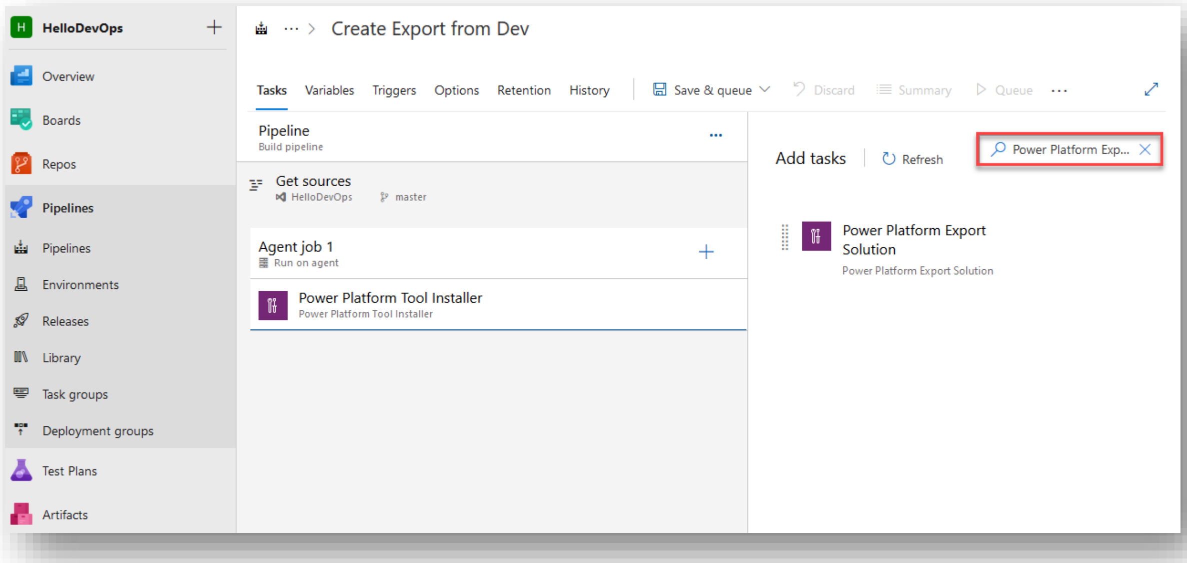
23. This will add the Tool Installer task to the Pipeline. No additional configuration is required for this task.



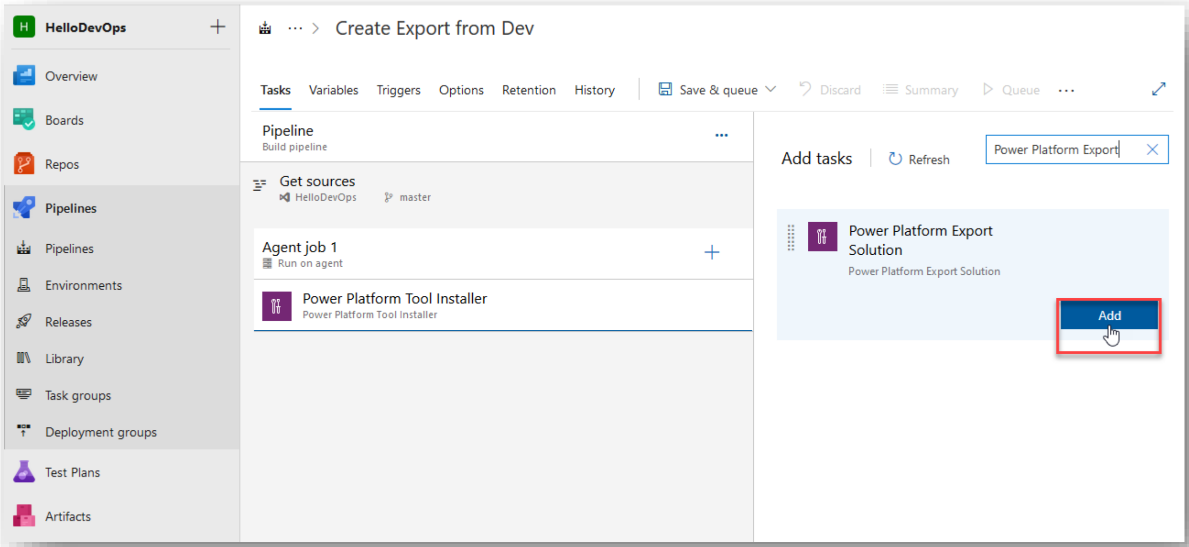
Export Solution as Unmanaged

Your unmanaged solution is your source code for your configuration. You should export your solution as unmanaged to check it into source control.

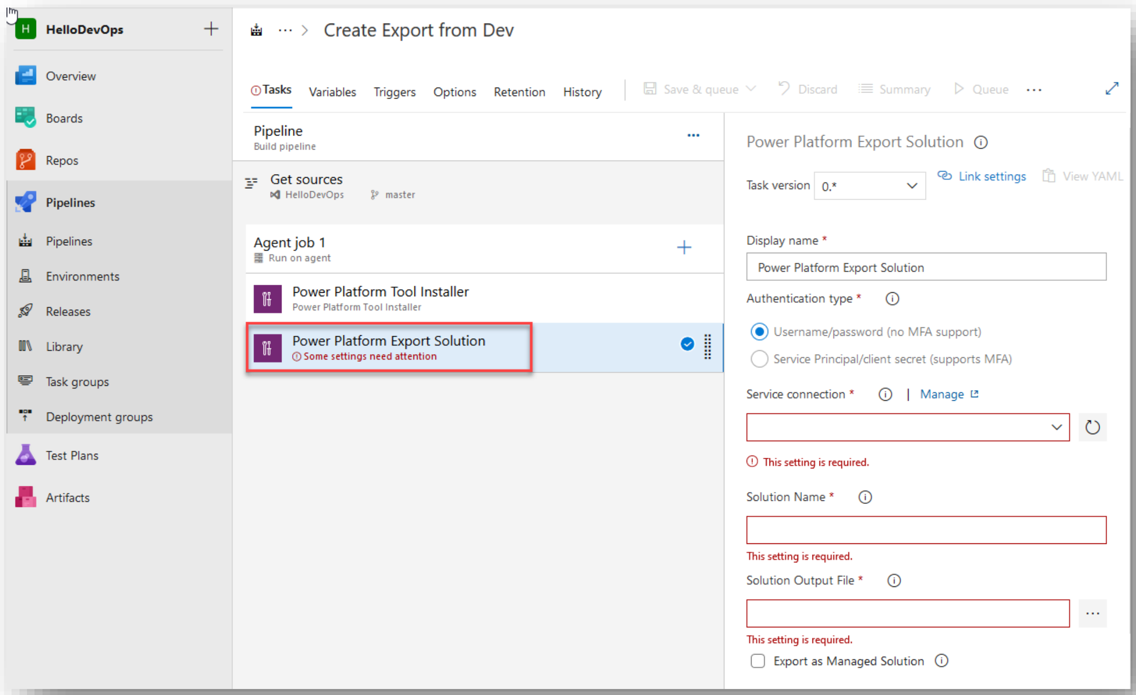
24. The next task you will add is a task that exports the unmanaged solution from development environment. Search for ‘Power Platform Export Solution’.



25. Select the task and Click ‘Add’.



26. After adding the Power Platform Export Solution task, you will notice that additional configuration is required. Click on the Power Platform Export Solution (in the pipeline view).



Power Platform Export Solution

Task version 0.*

Link settings | View YAML

Display name * Power Platform Export Solution

Authentication type *
- (●) Username/password (no MFA support)
- ( ) Service Principal/client secret (supports MFA)

Service connection * | Manage

This setting is required.

Solution Name *

This setting is required.

Solution Output File *

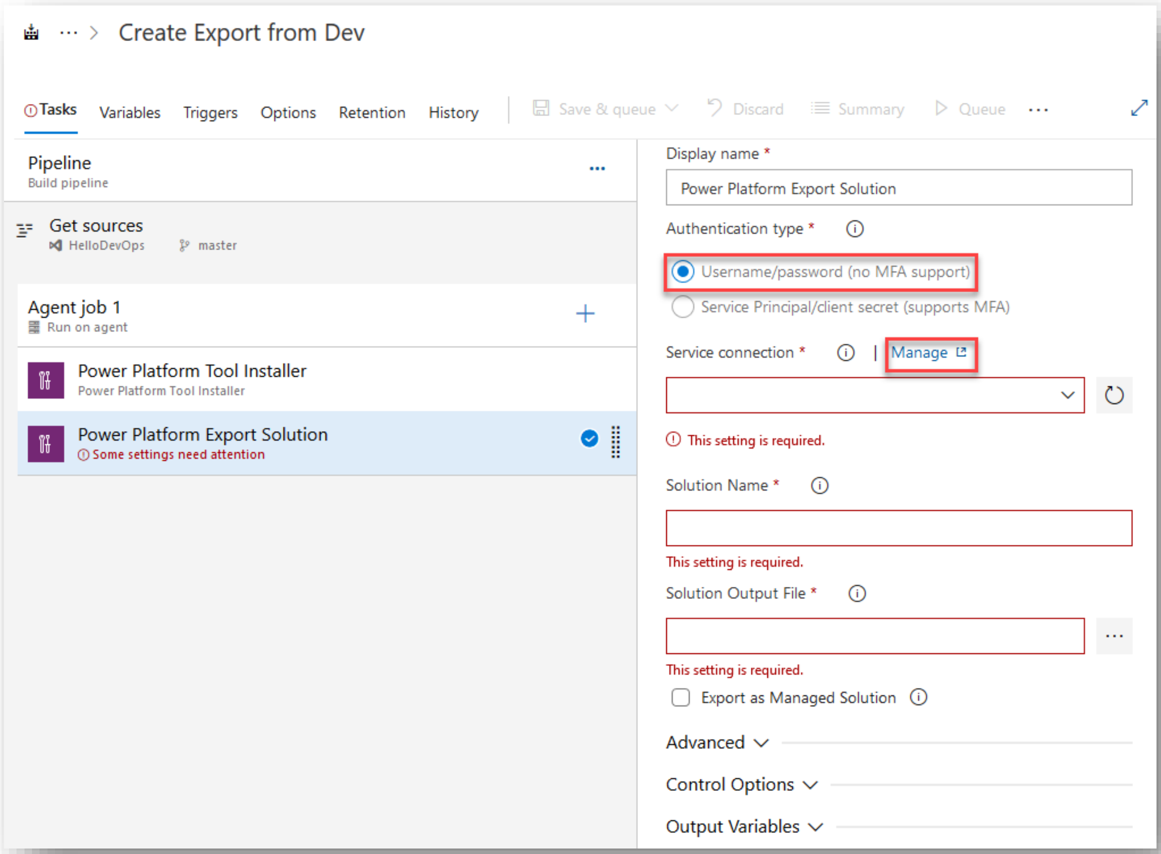
This setting is required.

☐ Export as Managed Solution

27. This will open the task configuration page:

- **Display name** is inherited from Build task itself.
- **Authentication type**: Two types of authentications are available:
    - **Username/password**: simple to setup but does not support multi factor authentication (MFA). This is what we will use for this lab.
    - **Service Principal/client secret**: recommended and supports MFA. This is harder to setup as it requires creation of the Service principal and client secret in the Azure Portal as well as creation of the application user in the Power Platform environment. Not used in this lab but anyone familiar with setting this up can use this option instead throughout the rest of the lab.
- **Service Connection**: This is the connection to the environment that you want to export the solution from. We will define this in the next step.
- **Solution Name**: This is the name of the solution you want to export.
- **Solution Output File**: This specifies the path and filename of the generated solution zip.
- **Export as Managed solution**: By default, solutions are exported as unmanaged (for development purposes.) Setting this flag exports the solution as Managed (used for deployment to any downstream environment such as Test, Pre-Prod and Production).

28. Select **Username/password** under ‘Authentication type’. Click on ‘Manage’ next to Service connection.



Create Export from Dev

Tasks | Variables | Triggers | Options | Retention | History | Save & queue | Discard | Summary | Queue

Pipeline
Build pipeline

Get sources
HelloDevOps master

Agent job 1
Run on agent

Power Platform Tool Installer
Power Platform Tool Installer

Power Platform Export Solution
Some settings need attention

Display name *
Power Platform Export Solution

Authentication type *
● Username/password (no MFA support)
○ Service Principal/client secret (supports MFA)

Service connection * | Manage

This setting is required.

Solution Name *

This setting is required.

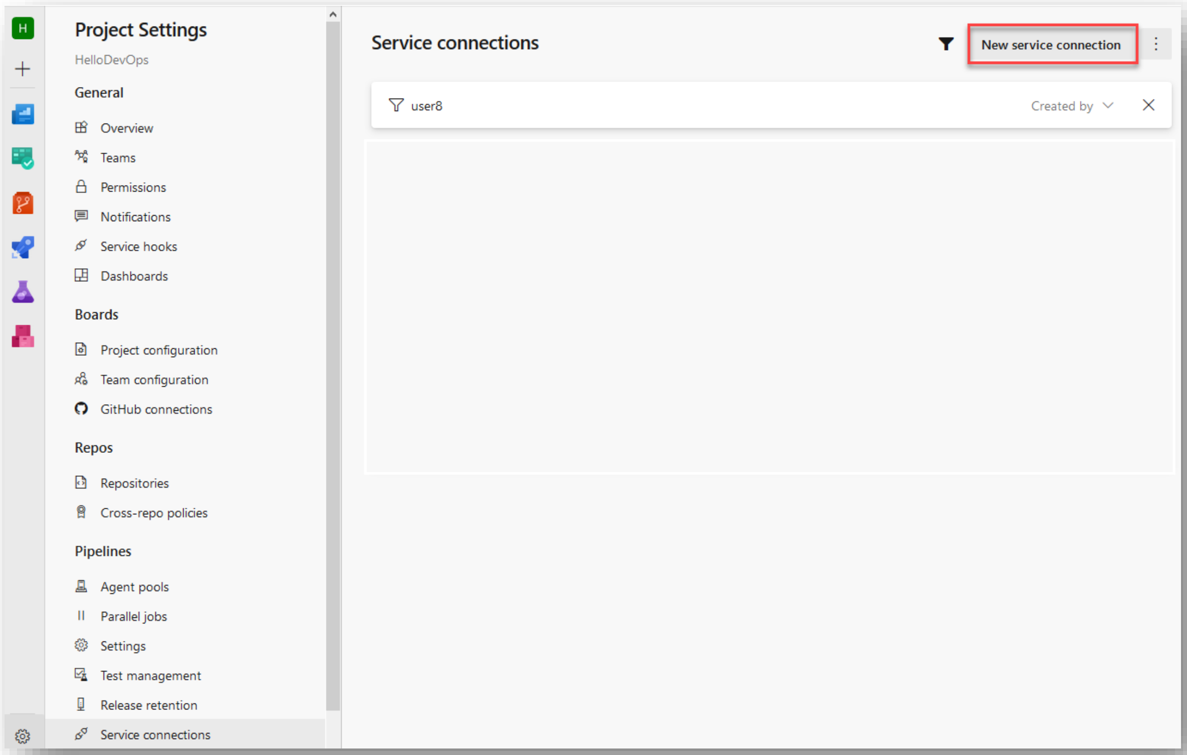
Solution Output File *

This setting is required.

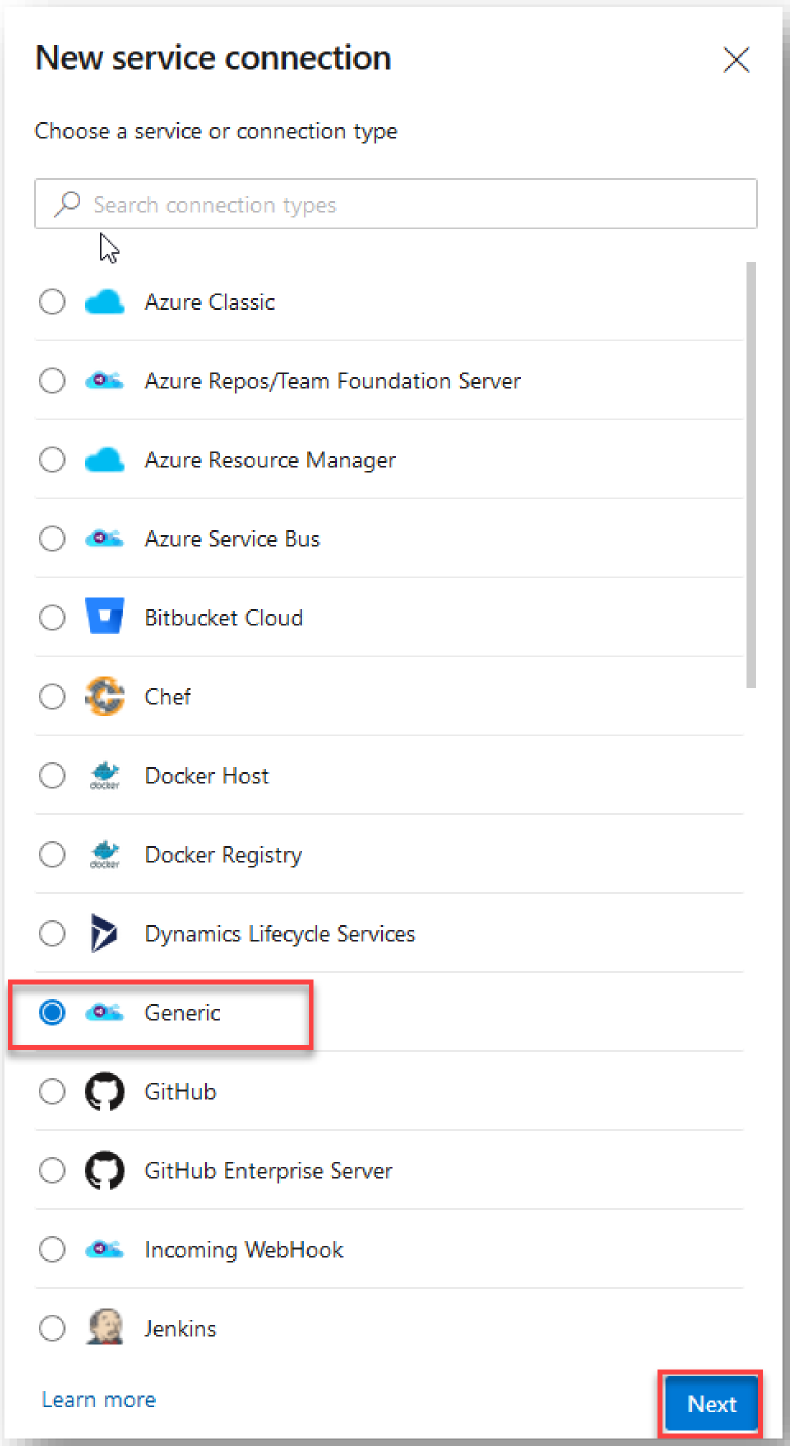
☐ Export as Managed Solution

Advanced
Control Options
Output Variables

29. This will bring up the connection configuration required to export the solution from the development environment **(user-xx-dev).** Click 'New service connection'.



30. This will bring up the New service connection screen. Select ‘Generic’ from the list of connections and click ‘Next’.

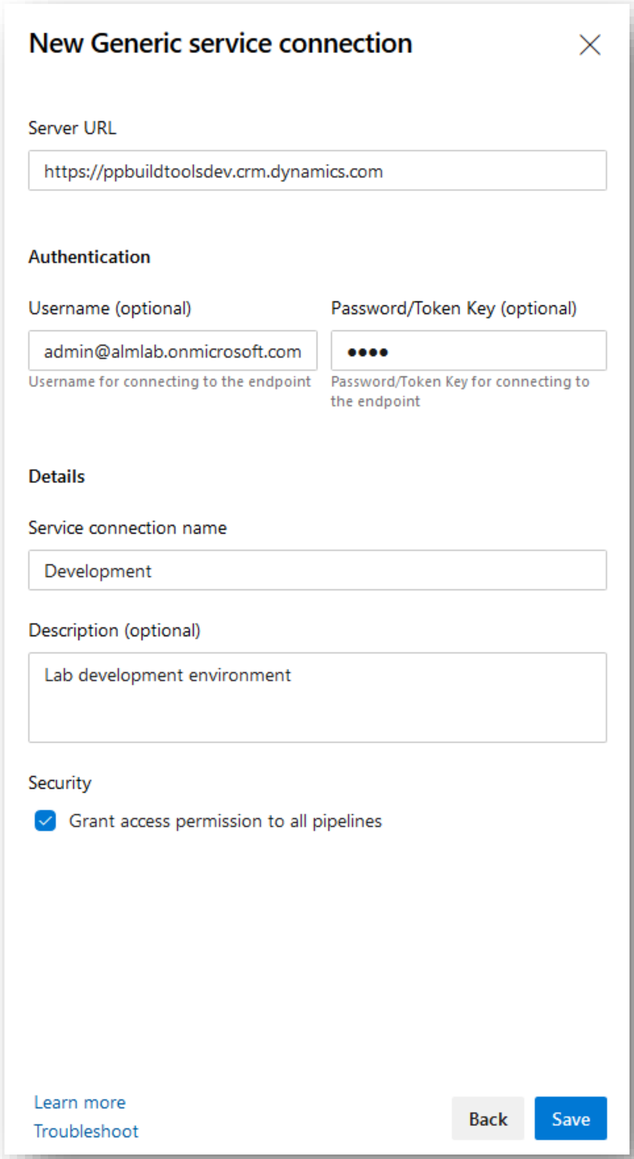


New service connection

Choose a service or connection type

31. Fill out the required details:
    - **Server URL**: https://<environment-url>.crm.dynamics.com. This should be the URL to your development environment.
    - **Username:** username of a user with admin access to the environment.
    - **Password:** Associated password for the user.
    - **Service Connection name:** This name will be used to identify which environment to connect to in the Build tasks.
    - **Description:** Give the connection a description that helps you identify the environment the service connection connects to.

    A sample connection is shown below.



New Generic service connection

Server URL: https://ppbuildtoolsdev.crm.dynamics.com

Authentication

Username (optional): admin@almlab.onmicrosoft.com
Username for connecting to the endpoint

Password/Token Key (optional): ●●●●
Password/Token Key for connecting to the endpoint

Details

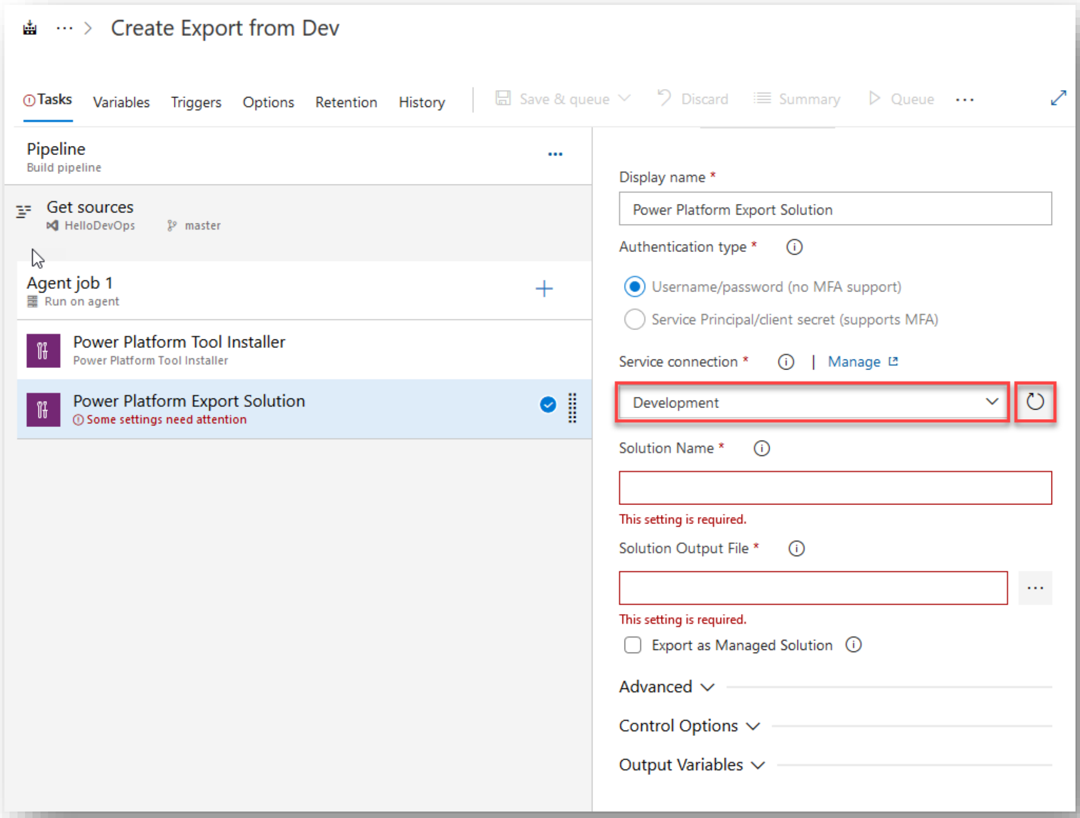
Service connection name: Development

Description (optional): Lab development environment

Security

✓ Grant access permission to all pipelines

32. Click ‘Save’. You will be in the pipeline service connections area in your project.
33. Close the browser tab and go back to the previous tab where you were building the pipeline. Select the Service connection that created in the previous step.
    Note: You might have to click the refresh button before the newly created Service connection is available to select .



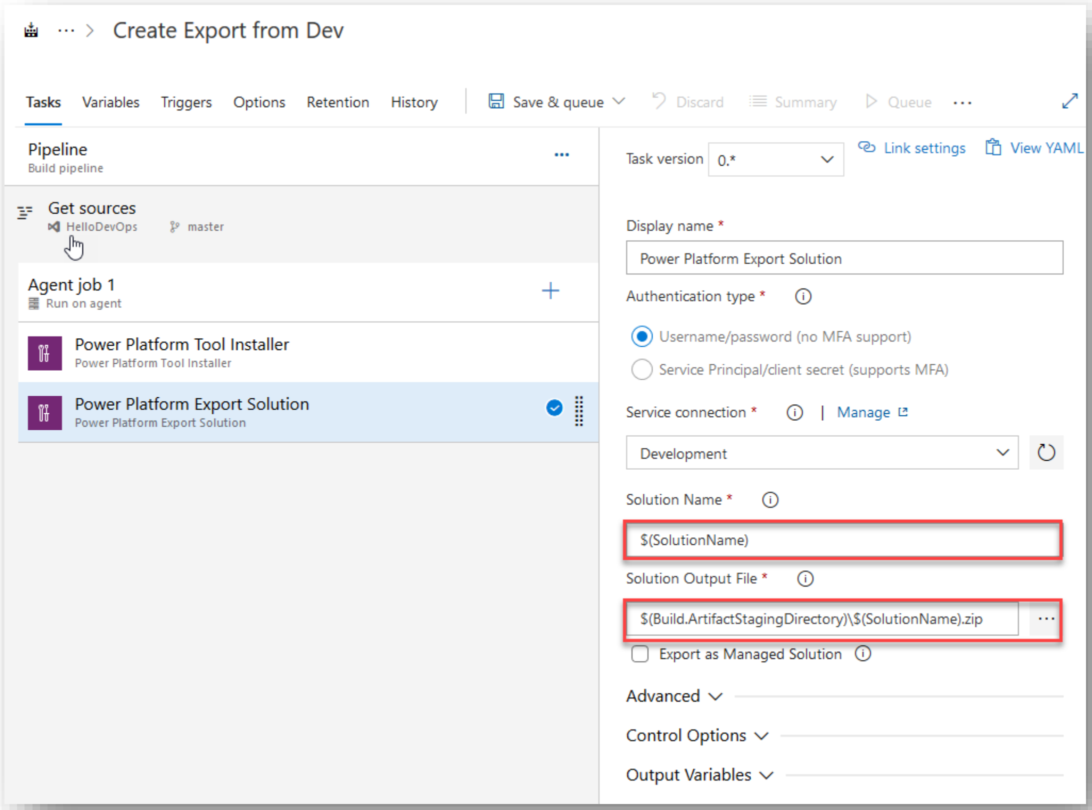
Create Export from Dev

Pipeline — Build pipeline
Get sources — HelloDevOps, master
Agent job 1 — Run on agent
Power Platform Tool Installer — Power Platform Tool Installer
Power Platform Export Solution — Some settings need attention

Display name *: Power Platform Export Solution
Authentication type *: [✓ Username/password (no MFA support)] Service Principal/client secret (supports MFA)
Service connection *: Development
Solution Name *: This setting is required.
Solution Output File *: This setting is required.
Export as Managed Solution: ☐

34. Fill out the remaining details:

- Solution Name: **$(SolutionName)**
  Note: This will use the input parameter that you specify when running (queuing) the build pipeline.
- Solution Output File: **$(Build.ArtifactStagingDirectory)\$(SolutionName).zip**
  Note: This will add the file to your repo and retain the existing solution name.



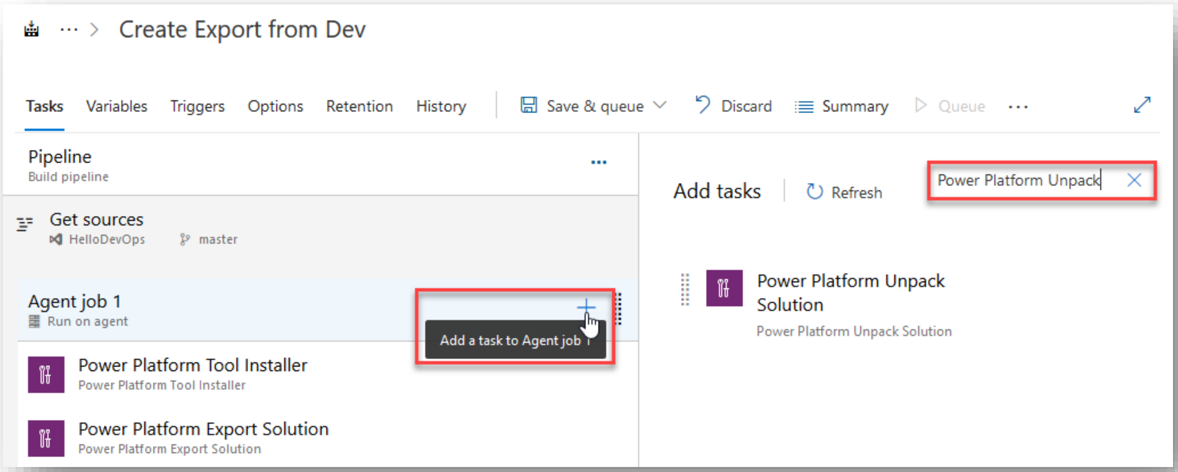
Task version: 0.*
Display name *: Power Platform Export Solution
Authentication type *: (●) Username/password (no MFA support) ( ) Service Principal/client secret (supports MFA)
Service connection * | Manage: Development
Solution Name *: $(SolutionName)
Solution Output File *: $(Build.ArtifactStagingDirectory)\$(SolutionName).zip
☐ Export as Managed Solution

35. Save the Build Pipeline by clicking Save & Queue, then save from the top command bar and clicking Save on the dialog.

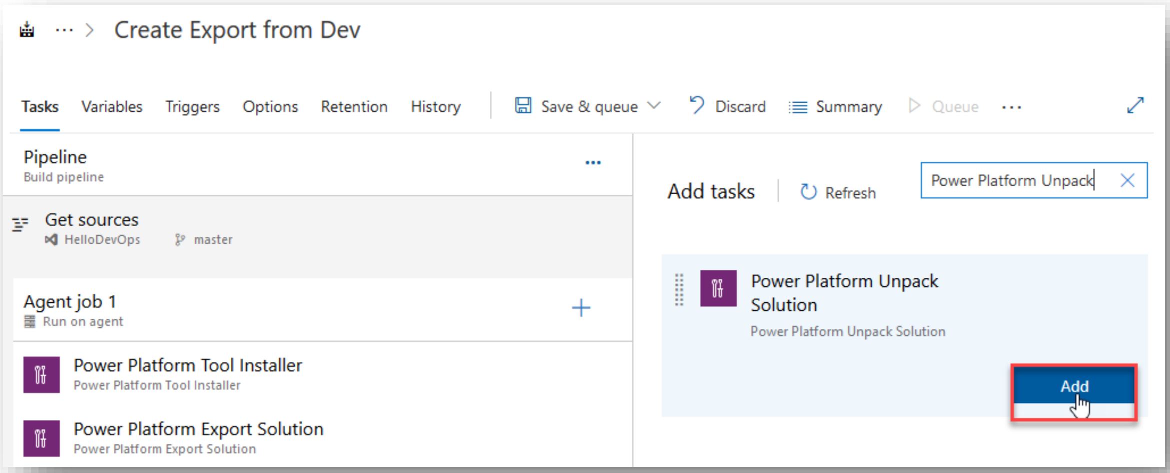
Unpack Solution

The solution file that is exported from the server is a zip file with consolidated configuration files. These initial files are not suitable for source code management as they are not structured to make it feasible for source code management systems to properly do differencing on the files and capture the changes you want to commit to source control. You need to 'unpack' the solution files to make them suitable for source control storage and processing.

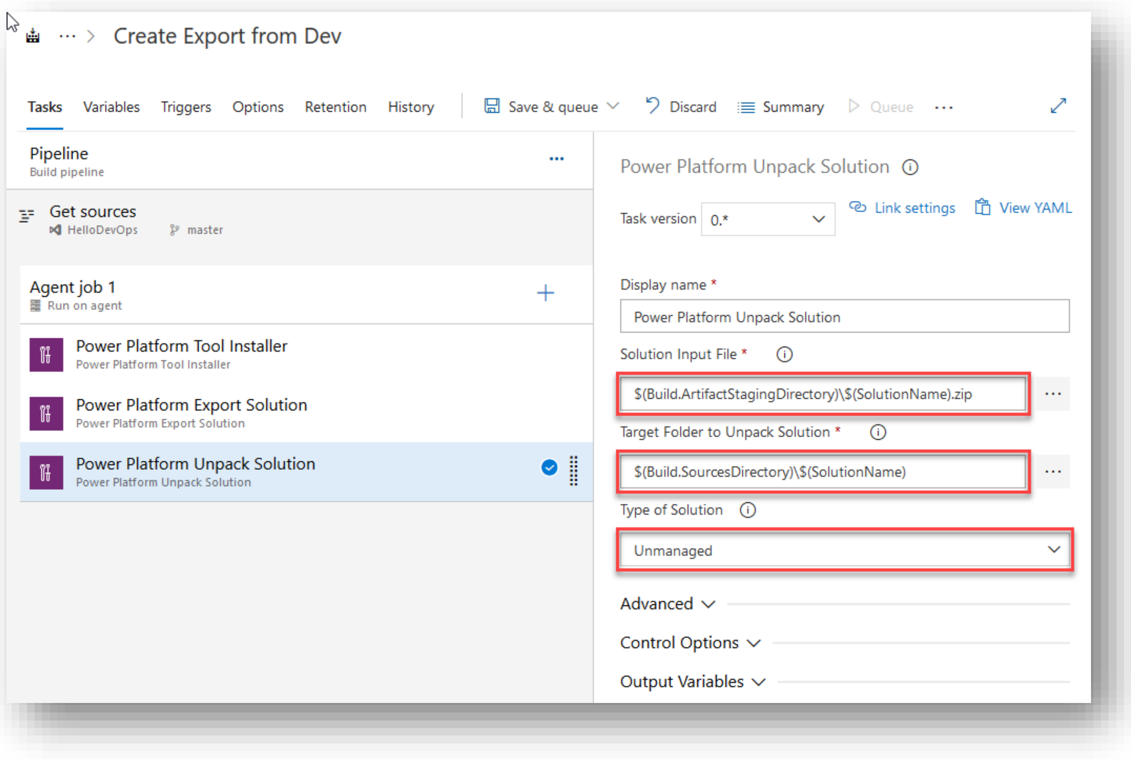
36. Click 'Add a task', then search for 'Power Platform Unpack'.



37. Add the Power Platform Unpack Solution task to the pipeline.



38. Open the task to configure settings for the task with the following details:
    - Solution Input File: $(Build.ArtifactStagingDirectory)\$(SolutionName).zip
    - Target Folder to unpack solution: $(Build.SourcesDirectory)\$(SolutionName)
    - Type of solution: Unmanaged



Power Platform Unpack Solution

Task version: 0.*

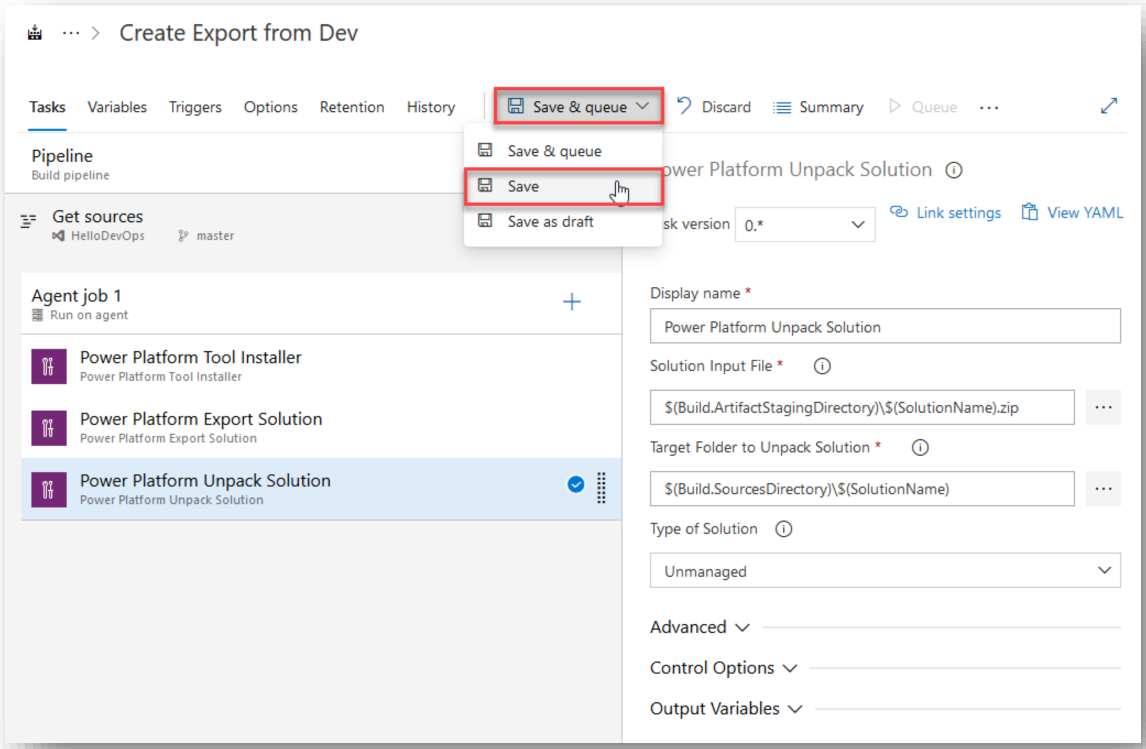
Display name *: Power Platform Unpack Solution

Solution Input File *: $(Build.ArtifactStagingDirectory)\$(SolutionName).zip

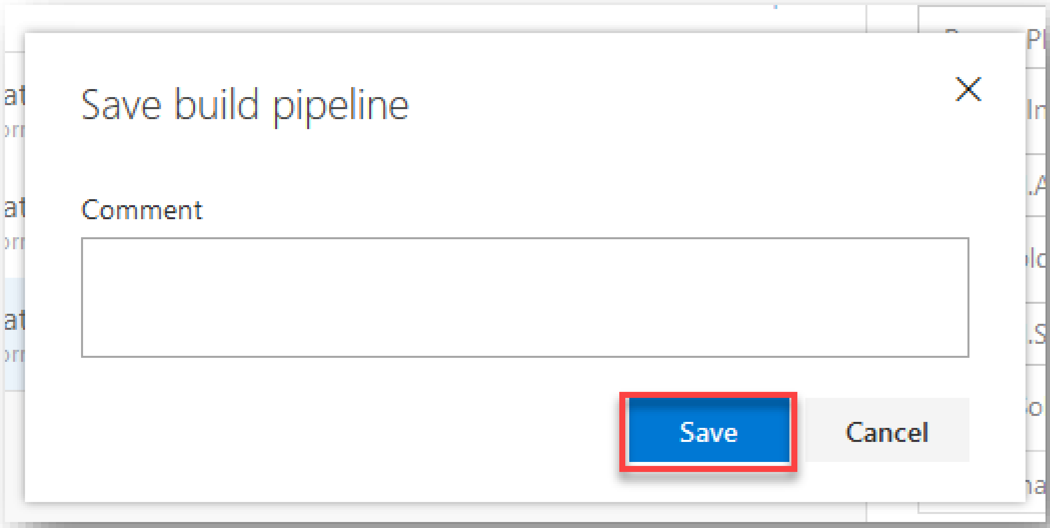
Target Folder to Unpack Solution *: $(Build.SourcesDirectory)\$(SolutionName)

Type of Solution: Unmanaged

39. Save the updated pipeline.


Create Export from Dev
Tasks
Variables
Triggers
Options
Retention
History
Save & queue
Discard
Summary
Queue
Save & queue
Save
Save as draft
Pipeline
Build pipeline
Get sources
HelloDevOps
master
Agent job 1
Run on agent
Power Platform Tool Installer
Power Platform Tool Installer
Power Platform Export Solution
Power Platform Export Solution
Power Platform Unpack Solution
Power Platform Unpack Solution
ower Platform Unpack Solution
sk version
0.*
Link settings
View YAML
Display name *
Power Platform Unpack Solution
Solution Input File *
$(Build.ArtifactStagingDirectory)\$(SolutionName).zip
Target Folder to Unpack Solution *
$(Build.SourcesDirectory)\$(SolutionName)
Type of Solution
Unmanaged
Advanced
Control Options
Output Variables

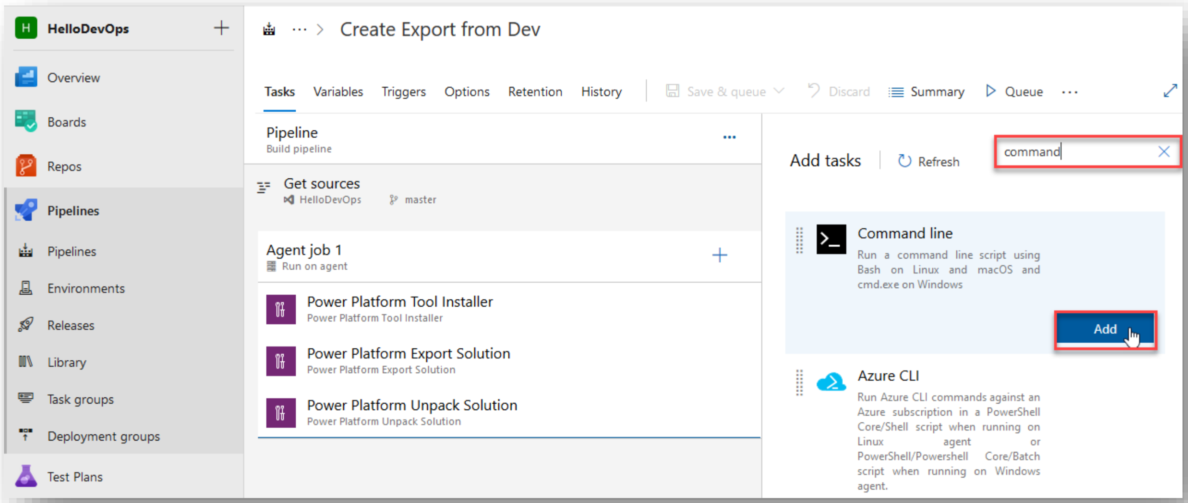
40. You can leave the comment blank.



Commit Solution Files to Source Control

Next, we are going to add some scripts as a next pipeline step to commit the solution to the repo. The reason we allowed for scripts to access the OAuth token when we started building the pipeline was to allow for this next step.

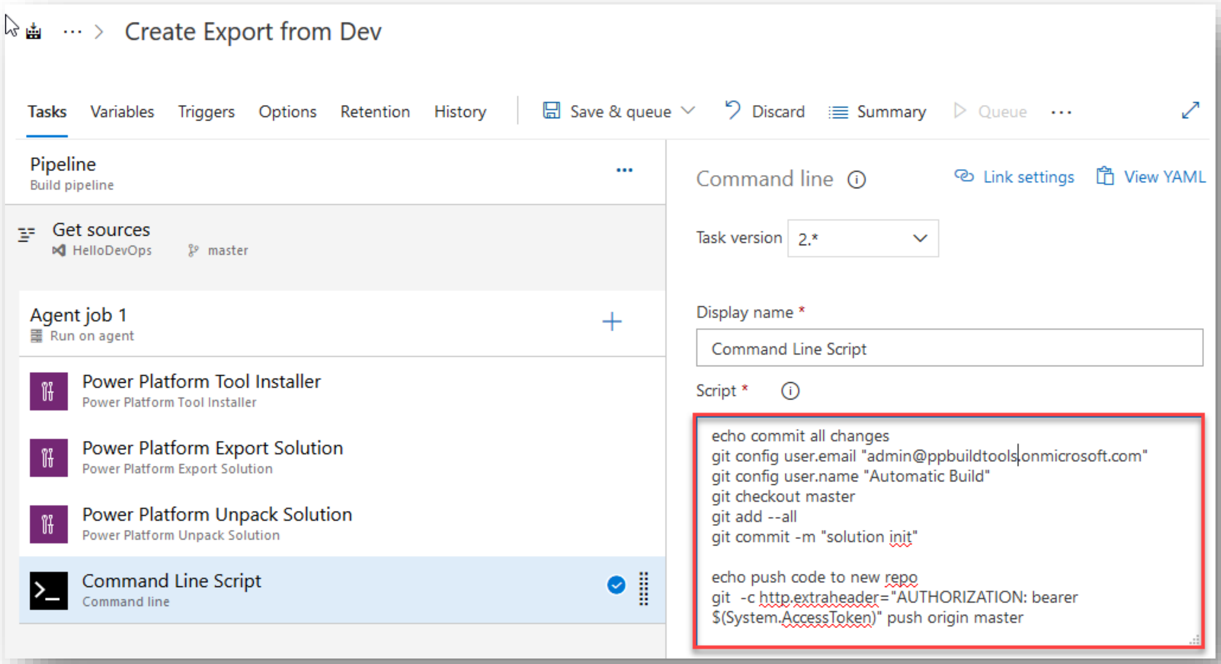
41. Add a Command Line task to your pipeline by clicking the + button, searching for 'command', and adding it.


HelloDevOps
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Releases
Library
Task groups
Deployment groups
Test Plans
Create Export from Dev
Tasks
Variables
Triggers
Options
Retention
History
Save & queue
Discard
Summary
Queue
Pipeline
Build pipeline
Get sources
HelloDevOps
master
Agent job 1
Run on agent
Power Platform Tool Installer
Power Platform Export Solution
Power Platform Unpack Solution
Add tasks
Refresh
command
Command line
Run a command line script using Bash on Linux and macOS and cmd.exe on Windows
Add
Azure CLI
Run Azure CLI commands against an Azure subscription in a PowerShell Core/Shell script when running on Linux agent or PowerShell/Powershell Core/Batch script when running on Windows agent.

42. Select the task, give the task a display name and copy the following script into the script textbox:

```
echo commit all changes

git config user.email "userXXX@wrkdevops.onmicrosoft.com"

git config user.name "Automatic Build"

git checkout master

git add --all

git commit -m "solution init"


echo push code to new repo

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

43. Screenshot provided below:



Task version: 2.*

Display name *: Command Line Script

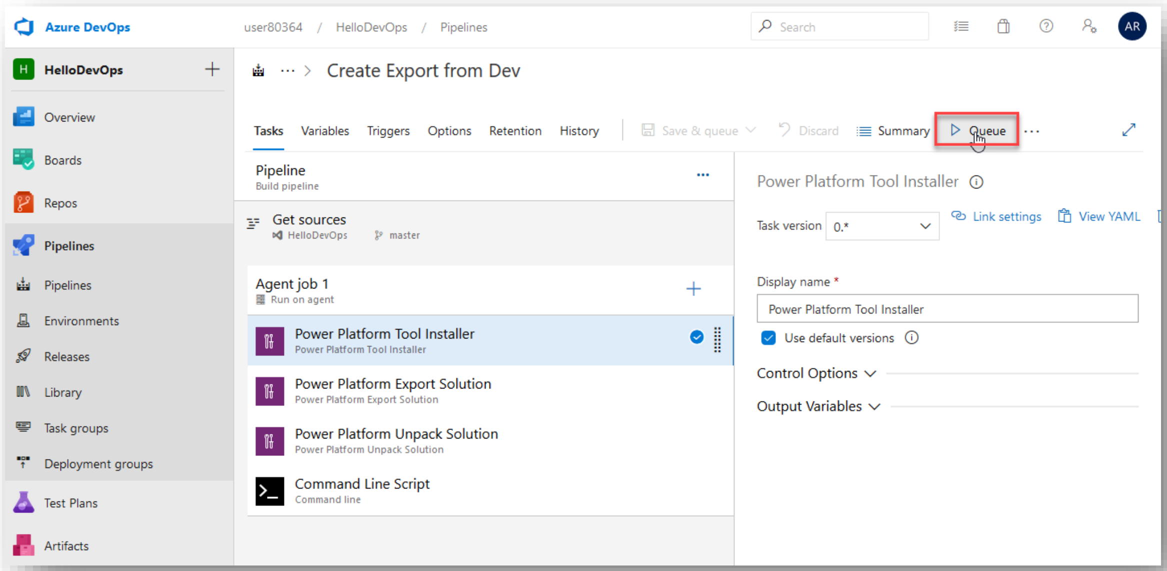
Script *:

```
echo commit all changes
git config user.email "admin@ppbuildtools.onmicrosoft.com"
git config user.name "Automatic Build"
git checkout master
git add --all
git commit -m "solution init"

echo push code to new repo
git  -c http.extraheader="AUTHORIZATION: bearer
$(System.AccessToken)" push origin master
```

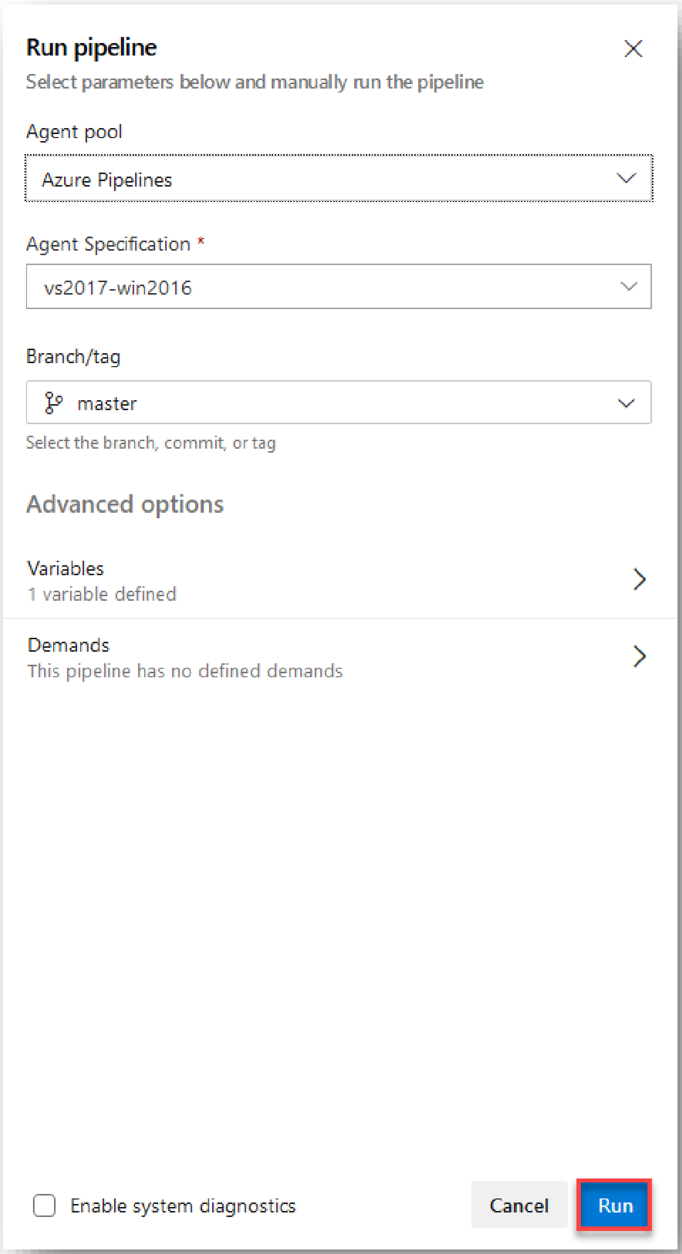
44. Save the task (Save & queue > Save).

Test your Pipeline

45. Select ‘Queue’ to execute your pipeline.



46. Leave the defaults and click ‘Run’.



Run pipeline

Select parameters below and manually run the pipeline

Agent pool: Azure Pipelines

Agent Specification *: vs2017-win2016

Branch/tag: master

Select the branch, commit, or tag

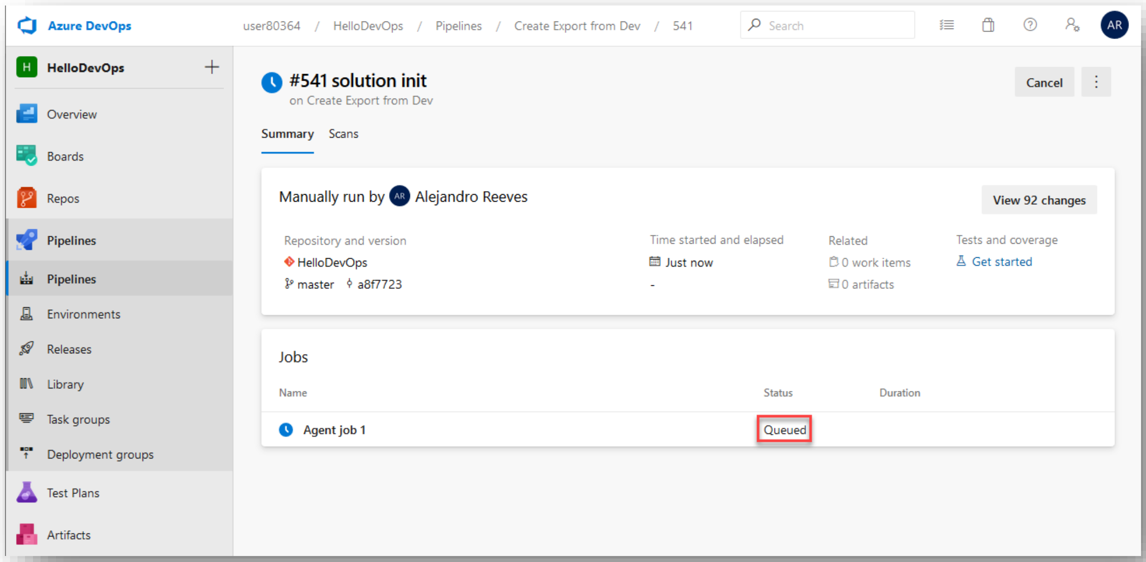
Advanced options

Variables
1 variable defined

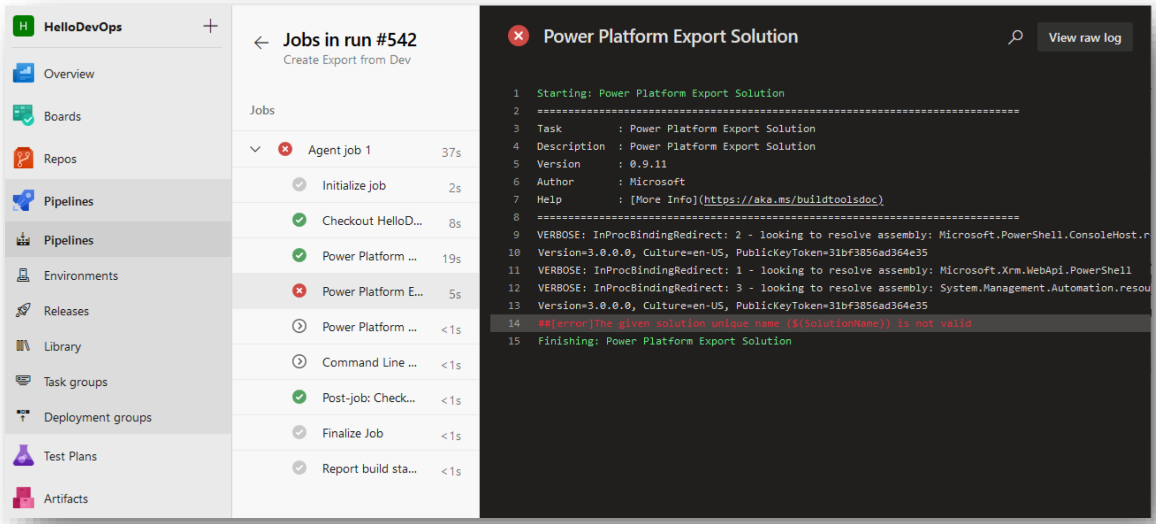
Demands
This pipeline has no defined demands

Enable system diagnostics

47. You can see your build has been queued. You can navigate to it directly in the notification on this screen or using the Builds area in the left navigation.

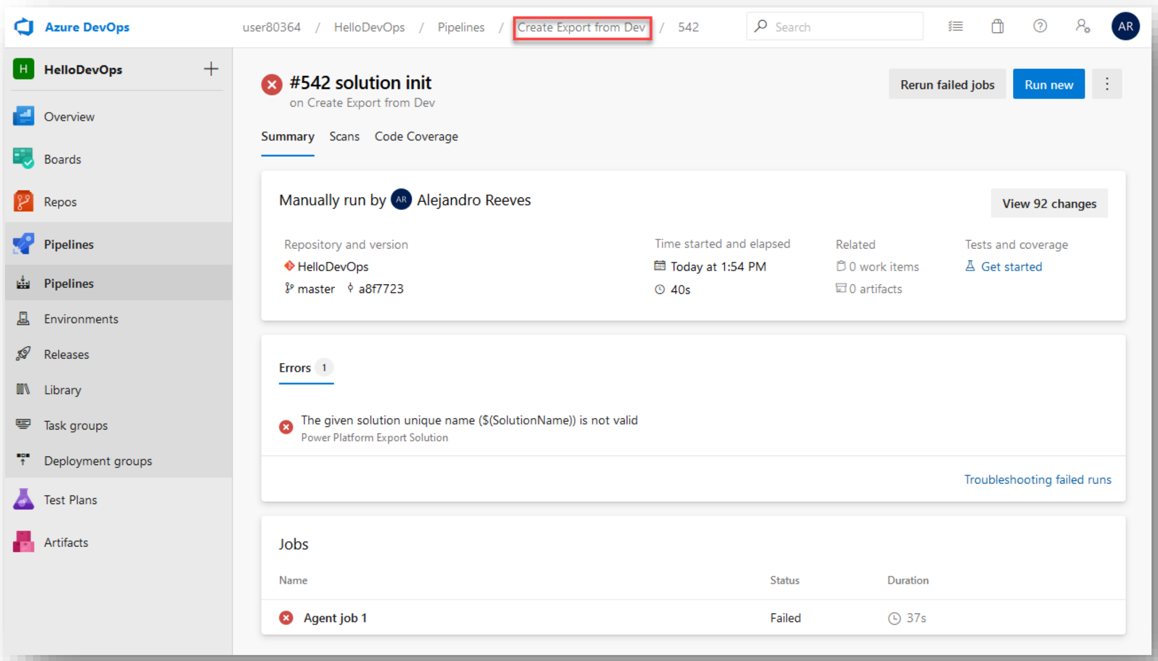


48. The desired outcome would be a series of green checkboxes, but if you have some issues with your pipeline, you should see the errors in the log. You can click on those for more details if you like. In our case, we have not defined our solution name variable yet, so the export step will fail. This was done intentionally to show you what a failure looks like and how to look at the details.



```
Starting: Power Platform Export Solution
==============================================================================
Task         : Power Platform Export Solution
Description  : Power Platform Export Solution
Version      : 0.9.11
Author       : Microsoft
Help         : [More Info](https://aka.ms/buildtoolsdoc)
==============================================================================
VERBOSE: InProcBindingRedirect: 2 - looking to resolve assembly: Microsoft.PowerShell.ConsoleHost.r
Version=3.0.0.0, Culture=en-US, PublicKeyToken=31bf3856ad364e35
VERBOSE: InProcBindingRedirect: 1 - looking to resolve assembly: Microsoft.Xrm.WebApi.PowerShell
VERBOSE: InProcBindingRedirect: 3 - looking to resolve assembly: System.Management.Automation.resou
Version=3.0.0.0, Culture=en-US, PublicKeyToken=31bf3856ad364e35
##[error]The given solution unique name ($(SolutionName)) is not valid
Finishing: Power Platform Export Solution
```

49. Go back to your pipeline by selecting the pipeline name from the top of screen. You also get there from clicking Pipelines in the left navigation and the select the pipeline you want to edit.



Manually run by Alejandro Reeves

Repository and version	Time started and elapsed	Related	Tests and coverage
HelloDevOps	Today at 1:54 PM	0 work items	Get started
master a8f7723	40s	0 artifacts	

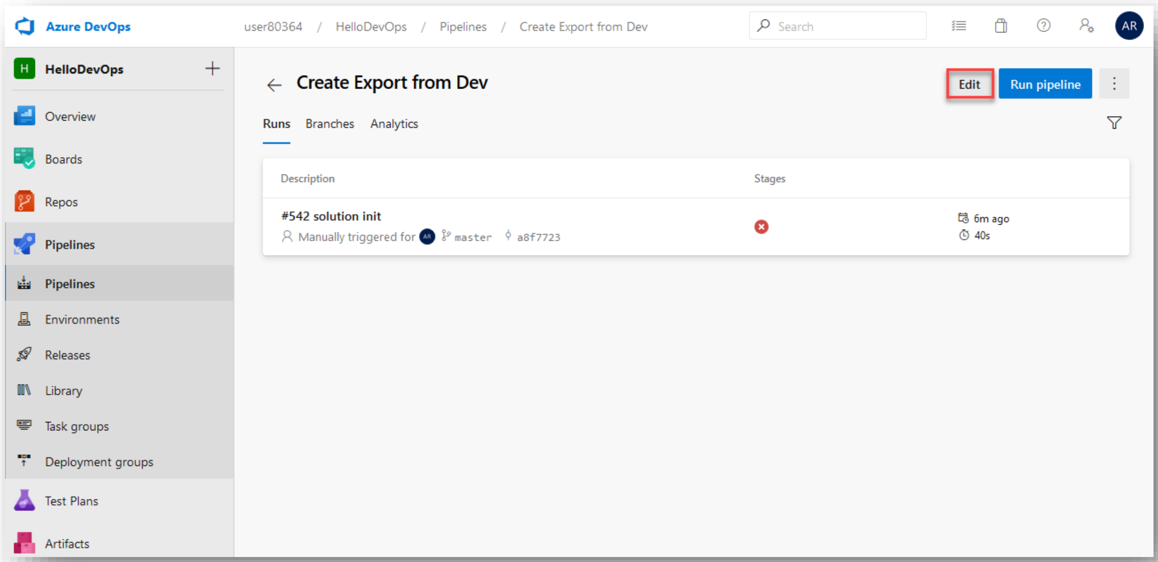
Errors 1

The given solution unique name ($(SolutionName)) is not valid
Power Platform Export Solution

Jobs

Name	Status	Duration
Agent job 1	Failed	37s

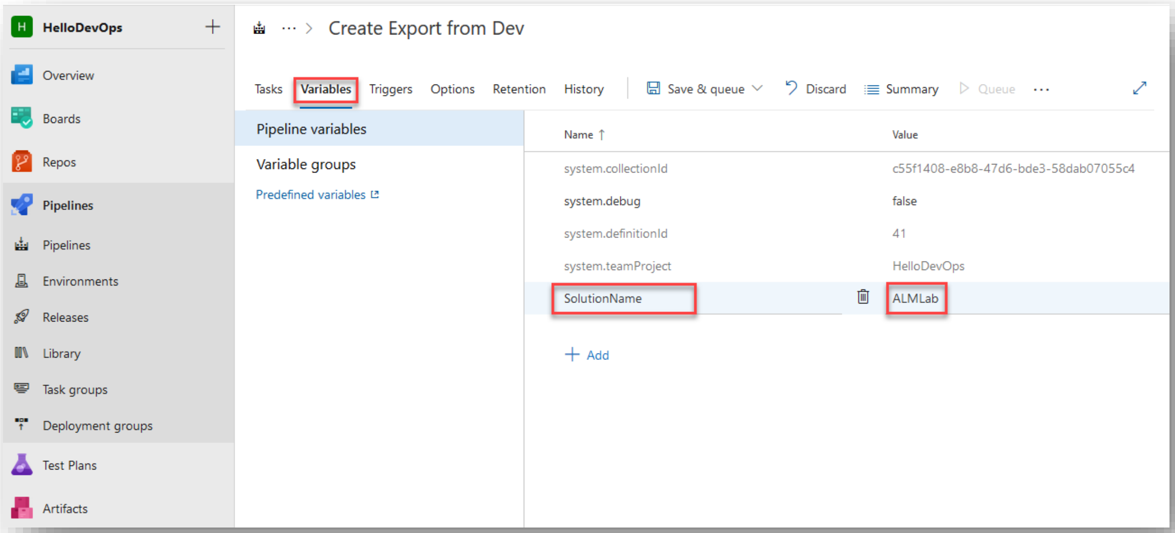
50. Click the 'Edit' button.



Description	Stages	
#542 solution init Manually triggered for master a8f7723		6m ago 40s

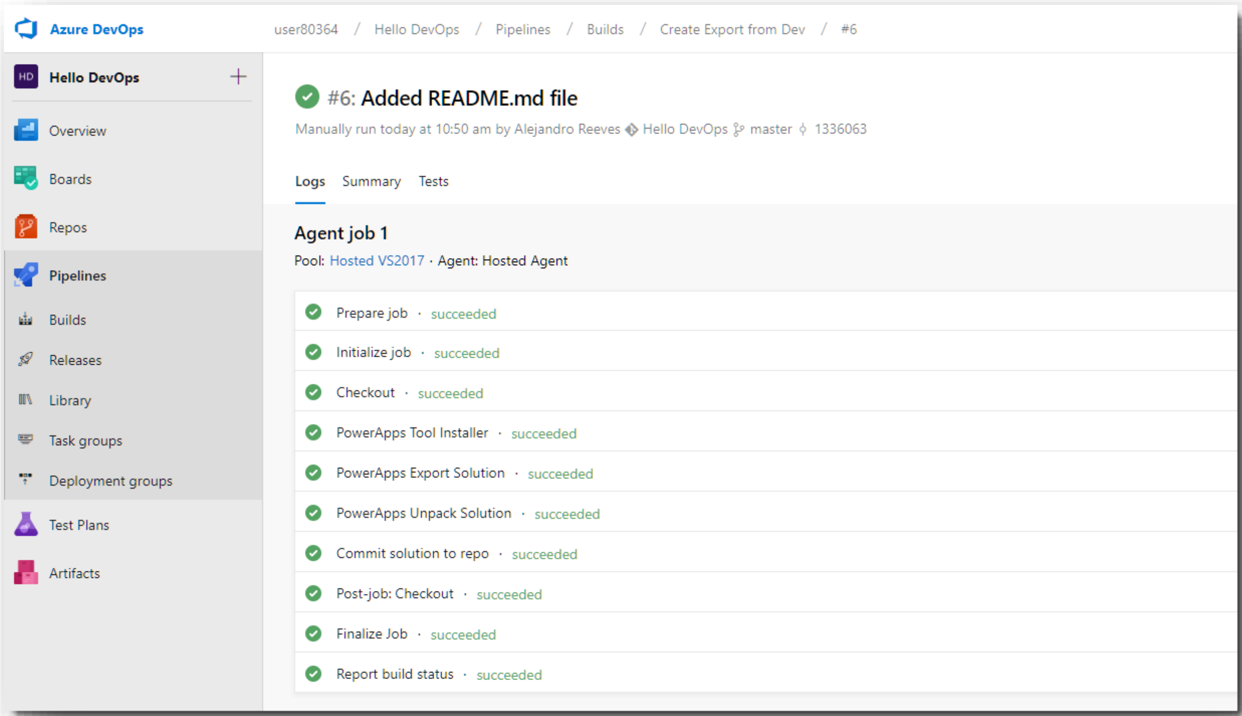
51. On the pipeline screen, switch to the Variables tab and click 'Add 'to add a new variable.

52. Use 'SolutionName' for the name of the variable and add your unique solution name as a value. It should be ALMLab if you are using the solution from Module 1.



Name ↑	Value
system.collectionId	c55f1408-e8b8-47d6-bde3-58dab07055c4
system.debug	false
system.definitionId	41
system.teamProject	HelloDevOps
SolutionName	ALMLab

53. Click Save and Queue and observe the results. Fix any issues until you get a successful build.



Azure DevOps | user80364 / Hello DevOps / Pipelines / Builds / Create Export from Dev / #6

Hello DevOps: Overview, Boards, Repos, Pipelines, Builds, Releases, Library, Task groups, Deployment groups, Test Plans, Artifacts

#6: Added README.md file

Manually run today at 10:50 am by Alejandro Reeves · Hello DevOps · master · 1336063

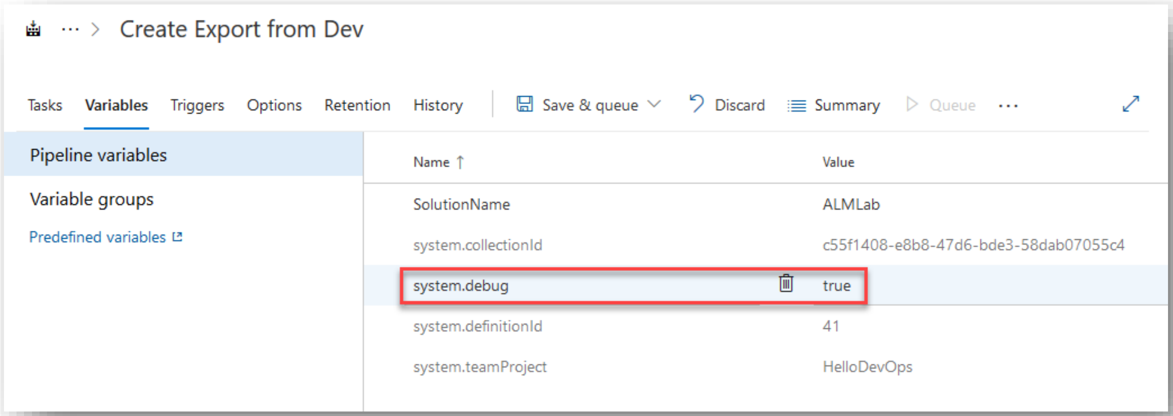
Logs | Summary | Tests

Agent job 1

Pool: Hosted VS2017 · Agent: Hosted Agent

```
Prepare job · succeeded
Initialize job · succeeded
Checkout · succeeded
PowerApps Tool Installer · succeeded
PowerApps Export Solution · succeeded
PowerApps Unpack Solution · succeeded
Commit solution to repo · succeeded
Post-job: Checkout · succeeded
Finalize Job · succeeded
Report build status · succeeded
```

Note: You can get more details about any issues found by setting the variable ‘debug’ to ‘true’.



Create Export from Dev

Tasks | Variables | Triggers | Options | Retention | History | Save & queue | Discard | Summary | Queue

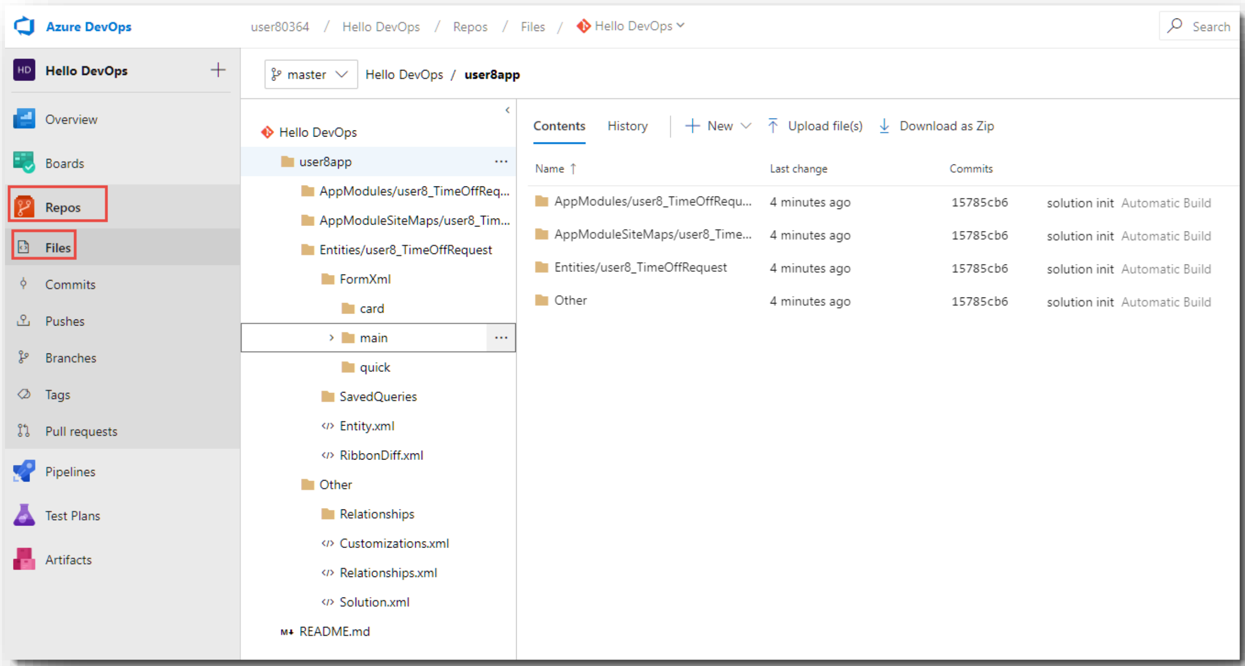
Pipeline variables

Variable groups

Predefined variables

Name	Value
SolutionName	ALMLab
system.collectionId	c55f1408-e8b8-47d6-bde3-58dab07055c4
system.debug	true
system.definitionId	41
system.teamProject	HelloDevOps

54. Let's go look at what it added to your source control now. Click Repos and Files and notice that it added a new folder that contains your solution files.

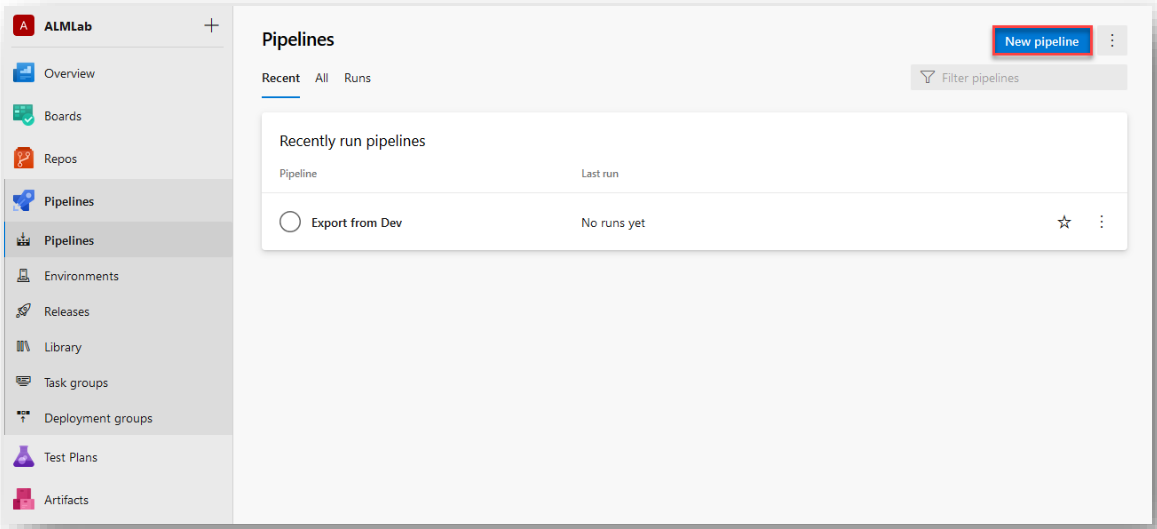


Name	Last change	Commits	
AppModules/user8_TimeOffRequ...	4 minutes ago	15785cb6	solution init Automatic Build
AppModuleSiteMaps/user8_Time...	4 minutes ago	15785cb6	solution init Automatic Build
Entities/user8_TimeOffRequest	4 minutes ago	15785cb6	solution init Automatic Build
Other	4 minutes ago	15785cb6	solution init Automatic Build

Pipeline 2: Build your Managed Solution

Your unmanaged solution is checked into source control, but you need to have a managed solution to deploy to other environments such as test and production. To obtain your managed solution, you should use a Just-In-Time (JIT) build environment where you would import your unmanaged solution files then export them as managed. These managed solution files will not be checked into source control but will be stored as a build artifact in your pipeline. This will make them available to be deployed in your release pipeline.

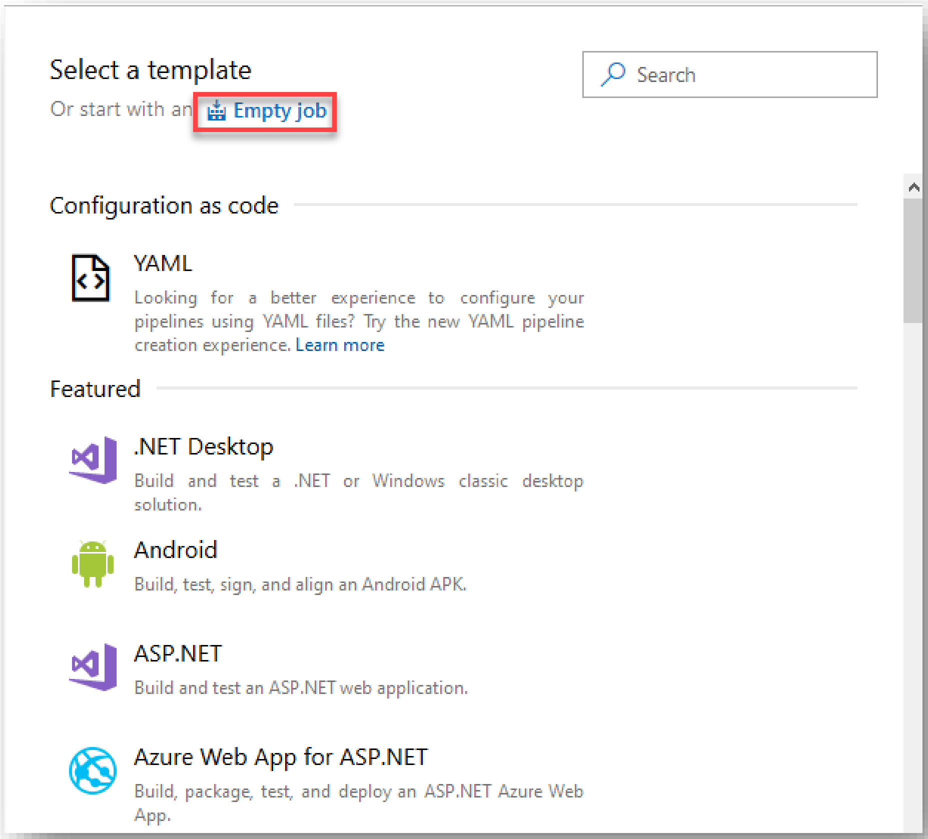
55. Navigate to your pipelines and add a new build pipeline.



Recently run pipelines

Pipeline	Last run
Export from Dev	No runs yet

56. Click “Use the classic editor” link to build the pipeline without YAML. On the next screen use the defaults and click ‘Continue’. Finally, click on the ‘Empty job’ link on the template selection page shown below.



Select a template

Or start with an Empty job

Search

Configuration as code

YAML

Looking for a better experience to configure your pipelines using YAML files? Try the new YAML pipeline creation experience. Learn more

Featured

.NET Desktop

Build and test a .NET or Windows classic desktop solution.

Android

Build, test, sign, and align an Android APK.

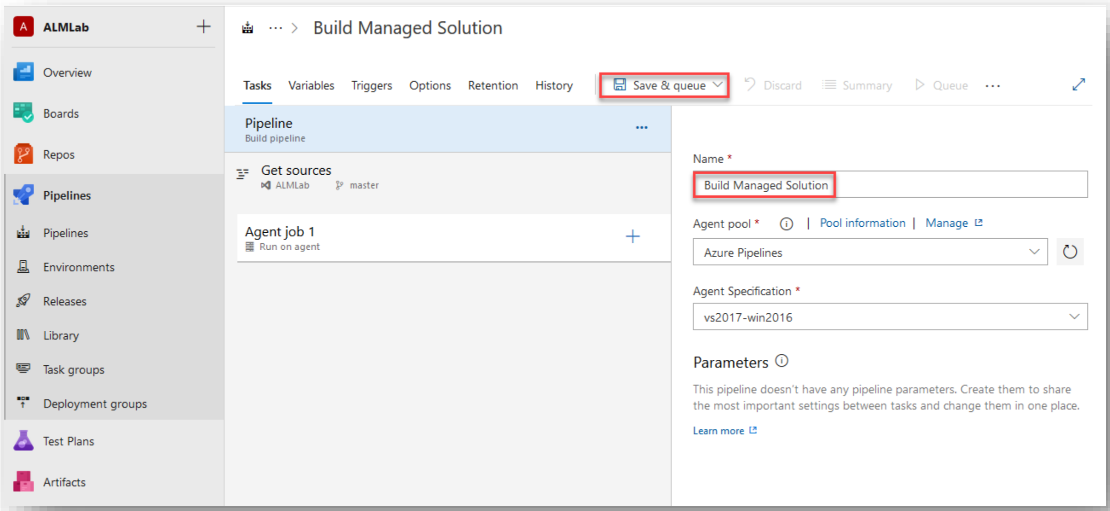
ASP.NET

Build and test an ASP.NET web application.

Azure Web App for ASP.NET

Build, package, test, and deploy an ASP.NET Azure Web App.

57. Name the pipeline 'Build Managed Solution' and Save it. This will give you an empty pipeline.



Name *
Build Managed Solution

Agent pool * | Pool information | Manage
Azure Pipelines

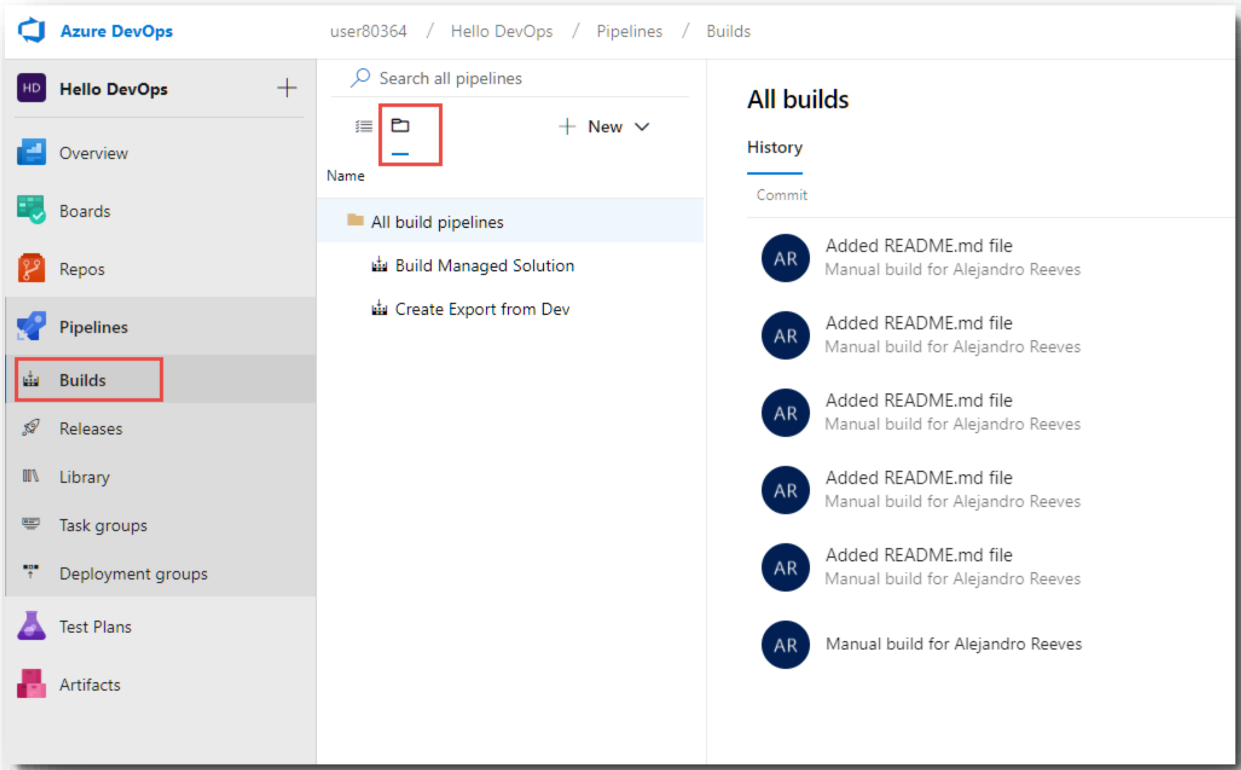
Agent Specification *
vs2017-win2016

Parameters

This pipeline doesn't have any pipeline parameters. Create them to share the most important settings between tasks and change them in one place.

Learn more

58. A small note on navigation. If you have problems seeing your newly created pipeline, you can click the little folder icon on the pipeline main page as the default view is only the recently used pipelines. If you have not used your pipeline yet and it does not show up in the list, change the view to all pipelines and you should be able to see and interact with it.



Added README.md file
Manual build for Alejandro Reeves

Added README.md file
Manual build for Alejandro Reeves

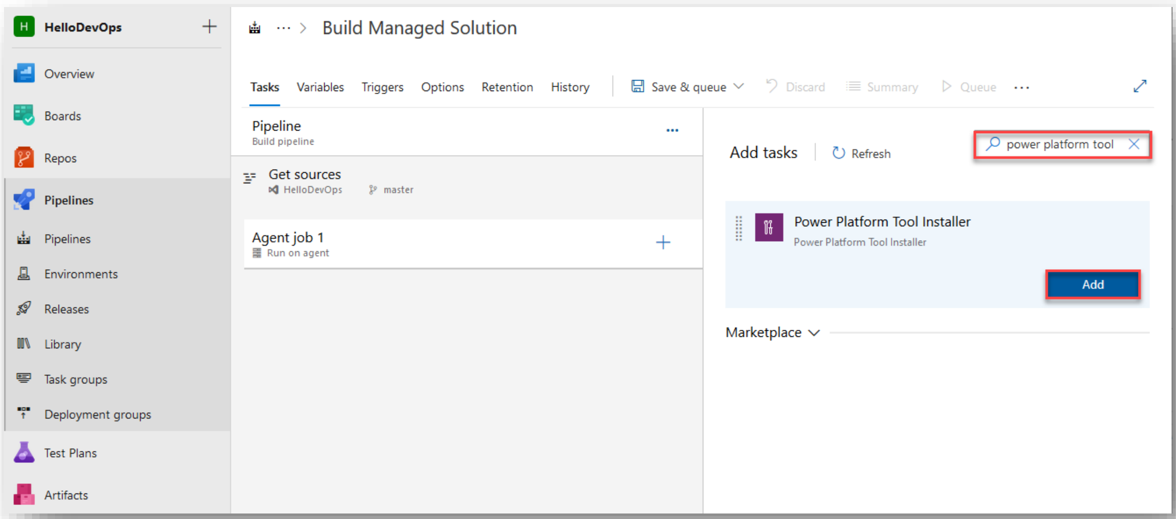
Added README.md file
Manual build for Alejandro Reeves

Added README.md file
Manual build for Alejandro Reeves

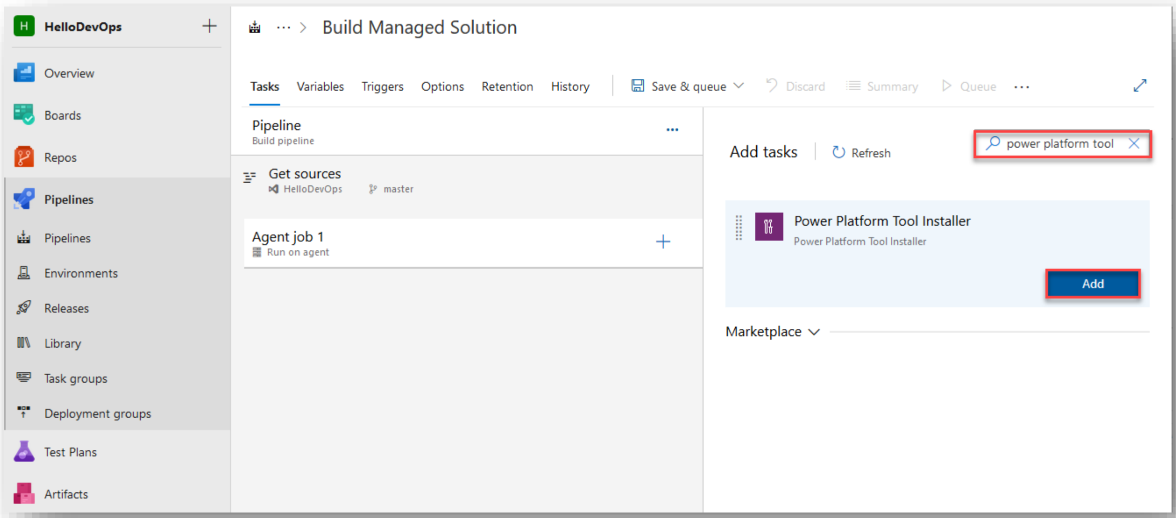
Added README.md file
Manual build for Alejandro Reeves

Manual build for Alejandro Reeves

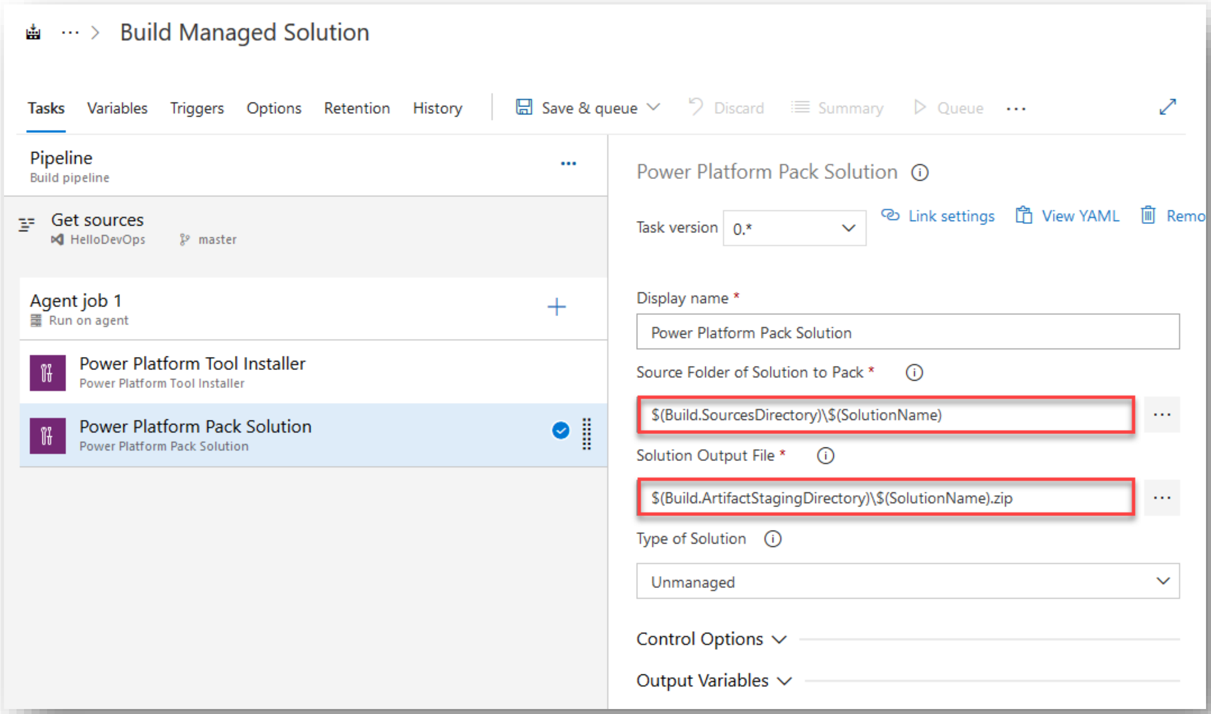
59. In your empty pipeline, add the Power Platform Tool Installer task.



60. Add the Power Platform Pack Solution task.



61. Configure the pack solution task with the following parameters:
    - Source Folder of Solution to Pack: $(Build.SourcesDirectory)\$(SolutionName)
    - Solution Output File: $(Build.ArtifactStagingDirectory)\$(SolutionName).zip



Power Platform Pack Solution

Task version: 0.*

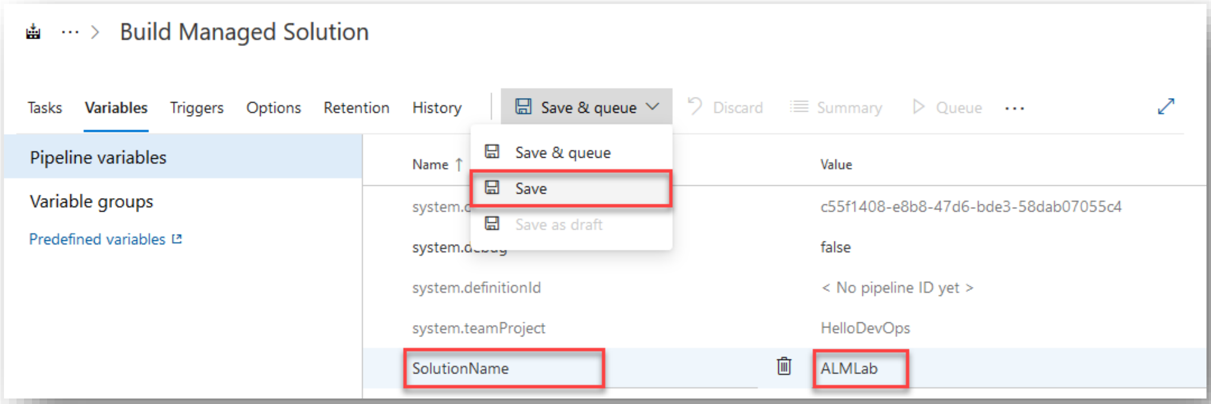
Display name *: Power Platform Pack Solution

Source Folder of Solution to Pack *: $(Build.SourcesDirectory)\$(SolutionName)

Solution Output File *: $(Build.ArtifactStagingDirectory)\$(SolutionName).zip

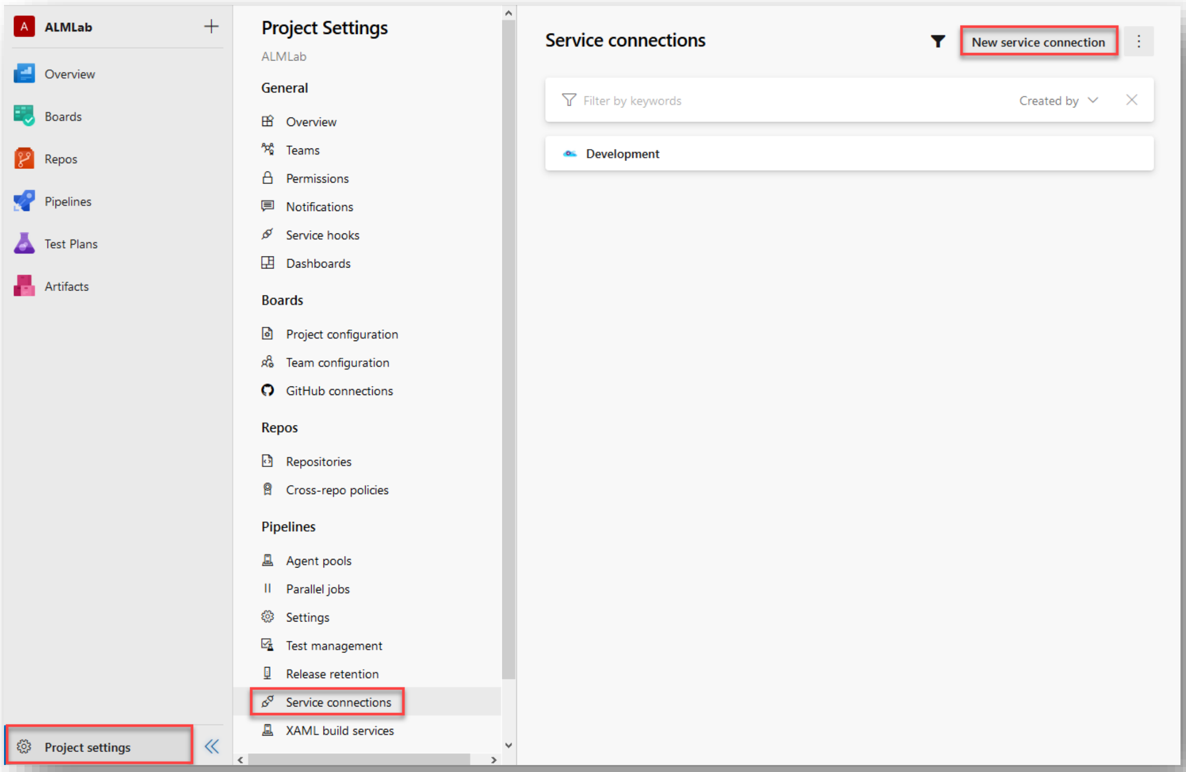
Type of Solution: Unmanaged

62. On the pipeline screen, switch to the variables tab in the pipeline and add the SolutionName variable with your unique solution name. This should be ALMLab if you are using the solution from Module 1. Click Save (through the Save & Queue dropdown).

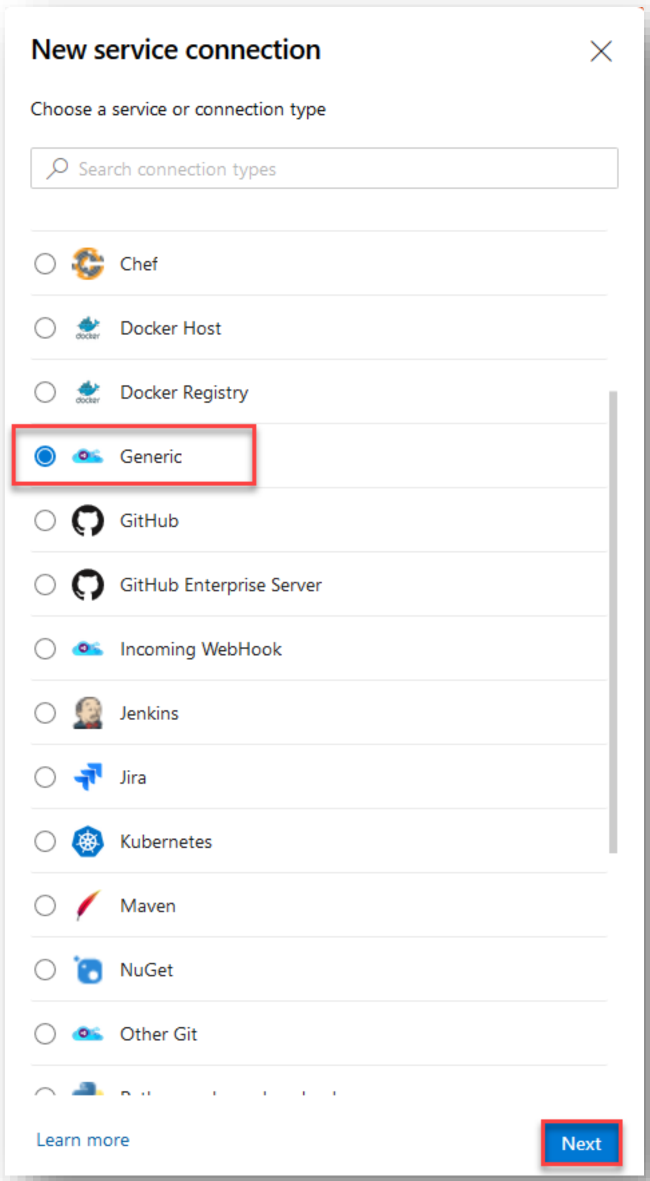


Name ↑	Value
system.c…	c55f1408-e8b8-47d6-bde3-58dab07055c4
system.debug	false
system.definitionId	< No pipeline ID yet >
system.teamProject	HelloDevOps
SolutionName	ALMLab

63. The next task will be to import the solution into your build server, so we need to add a connection to that environment before we add and configure the task to do the import. Click the 'Project settings' link in the lower left of the screen and navigate to 'Service connections', then click 'New service connection'.



64. Select ‘Generic’ on the New service connection page and click ‘Next’.


New service connection

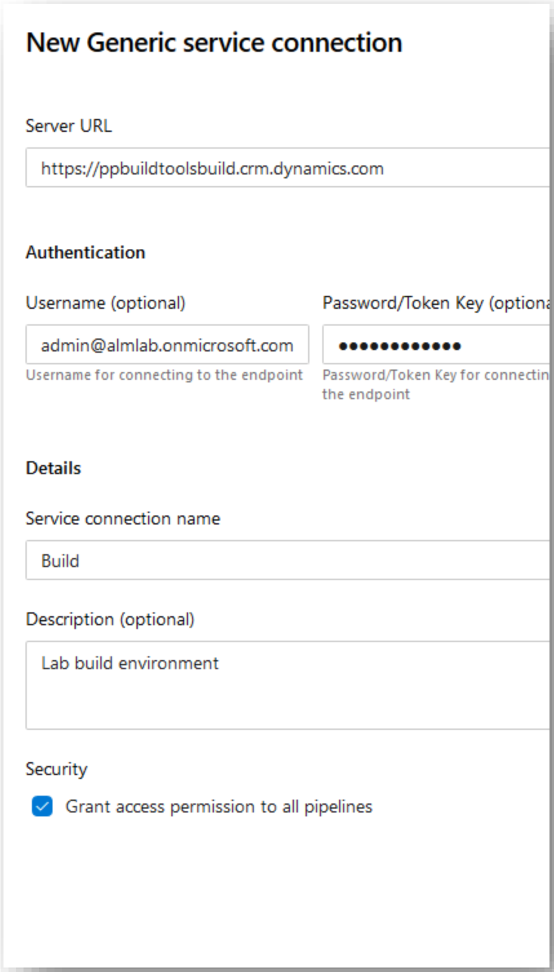
Choose a service or connection type

Search connection types

- Chef
- Docker Host
- Docker Registry
- [✓ Generic]
- GitHub
- GitHub Enterprise Server
- Incoming WebHook
- Jenkins
- Jira
- Kubernetes
- Maven
- NuGet
- Other Git

65. Fill out the required details:
    - **Server URL**: https://<environment-url>.crm.dynamics.com. This should be the URL to your build environment
    - **Username:** Username of a user with administrator access to the environment
    - **Password:** Associated password for the user
    - **Service Connection name:** This name will be used to identify which environment to connect to in the Build tasks
    - **Description:** Give the connection a description that helps you identify the environment the service connection connects to

    The sample connection is shown below.



New Generic service connection

Server URL

https://ppbuildtoolsbuild.crm.dynamics.com

Authentication

Username (optional): admin@almlab.onmicrosoft.com

Username for connecting to the endpoint

Password/Token Key (optiona: ●●●●●●●●●●●●

Password/Token Key for connectin the endpoint

Details

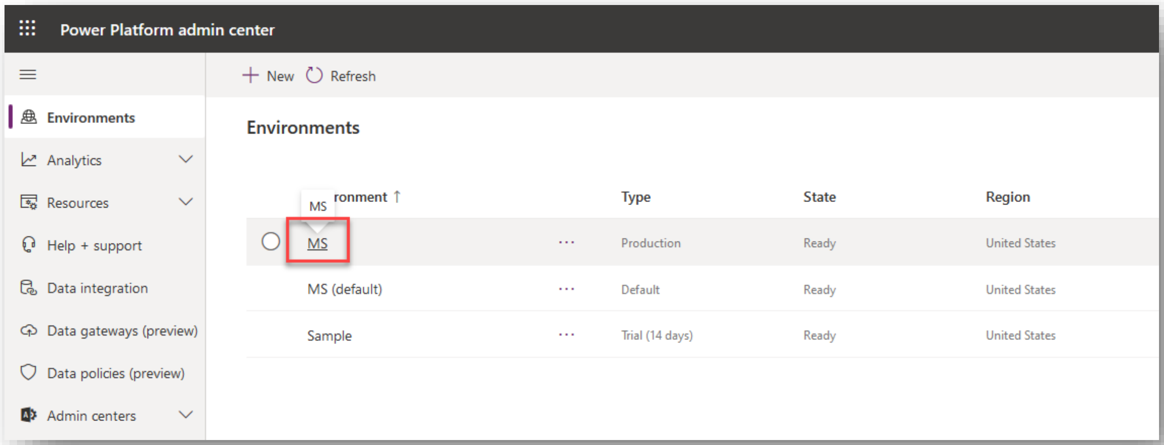
Service connection name: Build

Description (optional): Lab build environment

Security

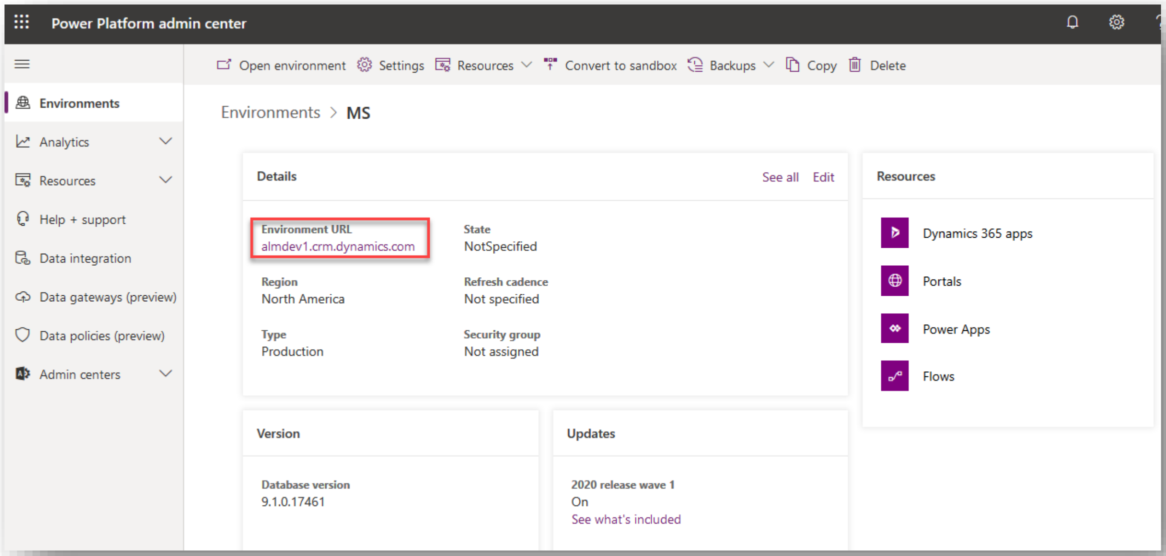
☑ Grant access permission to all pipelines

Note: If you forgot to record it the server URL for your build environment, simply visit https://admin.powerplatform.microsoft.com click the environment. This will open another displaying the environment URL. Make sure it is your build environment.



Environments

Environment ↑		Type	State	Region
MS	···	Production	Ready	United States
MS (default)	···	Default	Ready	United States
Sample	···	Trial (14 days)	Ready	United States



Environments > MS

Details

Environment URL
almdev1.crm.dynamics.com

State
NotSpecified

Region
North America

Refresh cadence
Not specified

Type
Production

Security group
Not assigned

Version

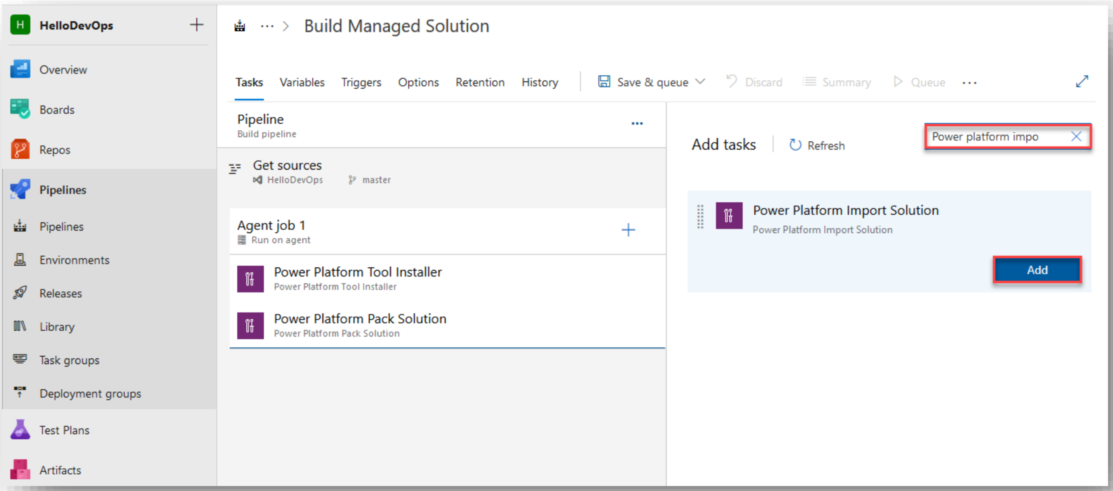
Database version
9.1.0.17461

Updates

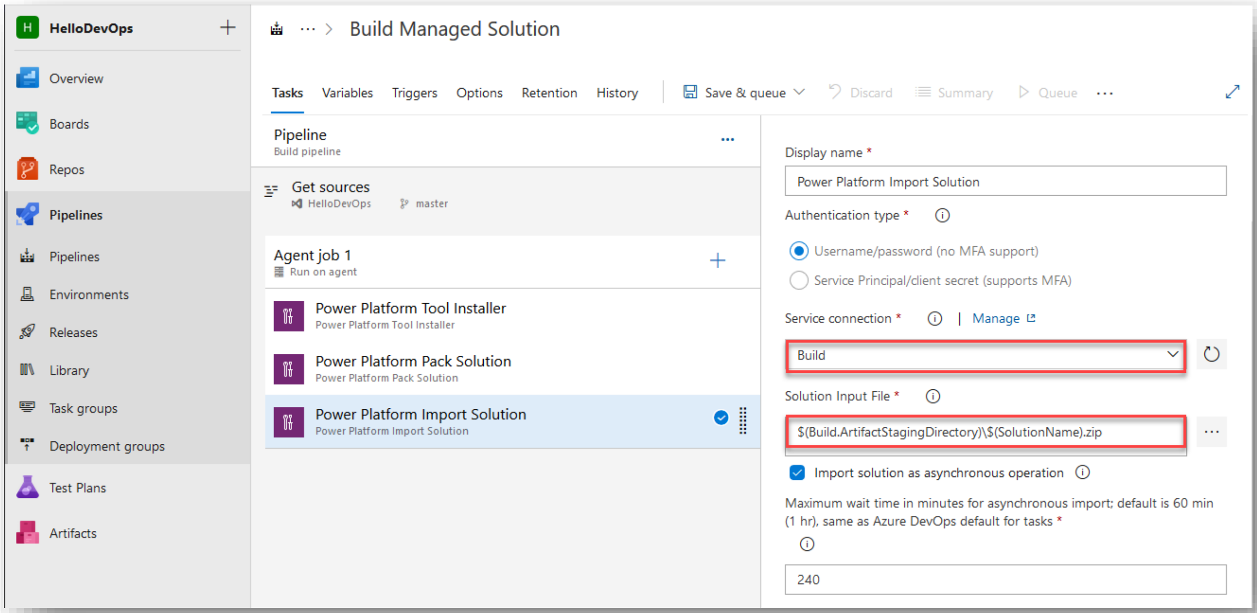
2020 release wave 1
On
See what's included

66. Click 'Save'. You will be in the pipeline service connections area in your project.

67. Navigate back to your Build Managed Solution pipeline (If you don't see the pipeline go to step 57 on how to view all pipelines). Click Edit, and then add the 'Power Platform Import Solution' task to take the solution file and import it into your build environment.



68. Select the connection you just added to your build server and provide the following for the solution input file: $(Build.ArtifactStagingDirectory)\$(SolutionName).zip.



Display name *
Power Platform Import Solution

Authentication type *
● Username/password (no MFA support)
○ Service Principal/client secret (supports MFA)

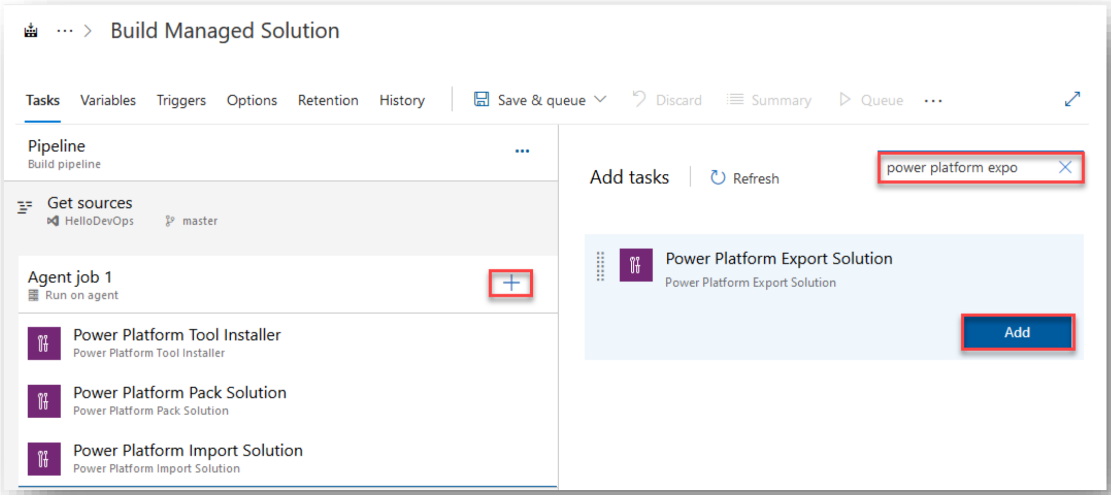
Service connection *
Build

Solution Input File *
$(Build.ArtifactStagingDirectory)\$(SolutionName).zip

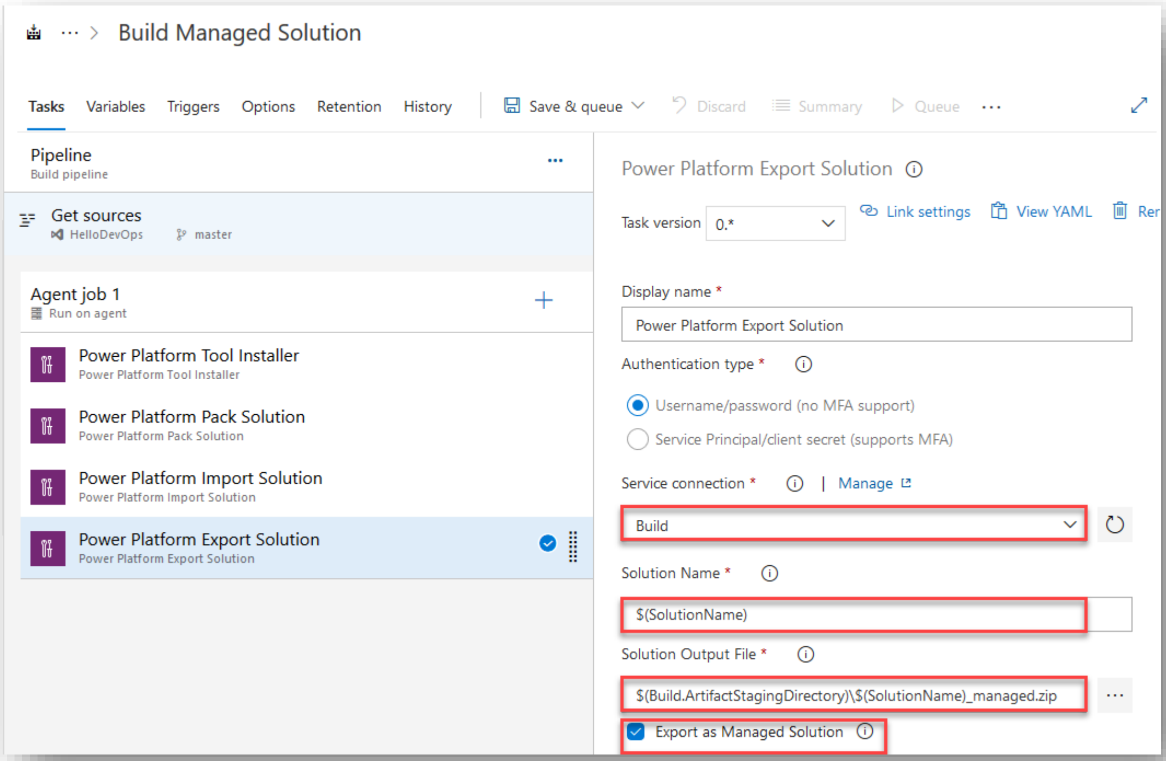
✓ Import solution as asynchronous operation

Maximum wait time in minutes for asynchronous import; default is 60 min (1 hr), same as Azure DevOps default for tasks *
240

69. Add a task to export the solution file as a managed file.


Build Managed Solution
Tasks
Variables
Triggers
Options
Retention
History
Save & queue
Discard
Summary
Queue
Pipeline
Build pipeline
Get sources
HelloDevOps
master
Agent job 1
Run on agent
Power Platform Tool Installer
Power Platform Tool Installer
Power Platform Pack Solution
Power Platform Pack Solution
Power Platform Import Solution
Power Platform Import Solution
Add tasks
Refresh
power platform expo
Power Platform Export Solution
Power Platform Export Solution
Add

70. Select your build environment, check the ’Export as managed solution’ checkbox and enter the following details:
    Solution Name: $(SolutionName)
    Solution Output File: $(Build.ArtifactStagingDirectory)\$(SolutionName)_managed.zip


Build Managed Solution
Tasks
Variables
Triggers
Options
Retention
History
Save & queue
Discard
Summary
Queue
Pipeline
Build pipeline
Get sources
HelloDevOps
master
Agent job 1
Run on agent
Power Platform Tool Installer
Power Platform Tool Installer
Power Platform Pack Solution
Power Platform Pack Solution
Power Platform Import Solution
Power Platform Import Solution
Power Platform Export Solution
Power Platform Export Solution
Power Platform Export Solution
Link settings
View YAML

Task version: 0.*

Display name *: Power Platform Export Solution

Authentication type *: [✓ Username/password (no MFA support)] Service Principal/client secret (supports MFA)

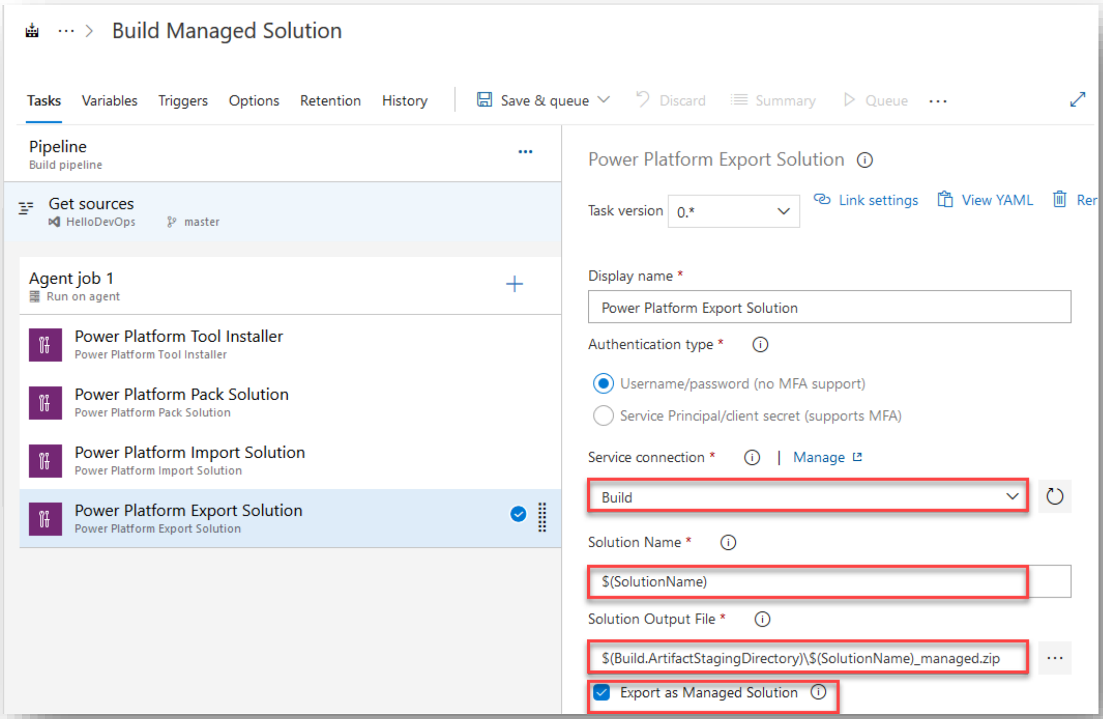
Service connection * | Manage: Build

Solution Name *: $(SolutionName)

Solution Output File *: $(Build.ArtifactStagingDirectory)\$(SolutionName)_managed.zip

[✓] Export as Managed Solution

71. The files at this stage will be in the build agent and we need to publish it as a build artifact. Add a new pipeline step Publish Pipeline Artifact.


Build Managed Solution
Tasks
Variables
Triggers
Options
Retention
History
Save & queue
Discard
Summary
Queue
Pipeline
Build pipeline
Get sources
HelloDevOps
master
Agent job 1
Run on agent
Power Platform Tool Installer
Power Platform Tool Installer
Power Platform Pack Solution
Power Platform Pack Solution
Power Platform Import Solution
Power Platform Import Solution
Power Platform Export Solution
Power Platform Export Solution
Power Platform Export Solution
Link settings
View YAML
Rer

Task version: 0.*

Display name *: Power Platform Export Solution

Authentication type *
- [✓ Username/password (no MFA support)]
- Service Principal/client secret (supports MFA)

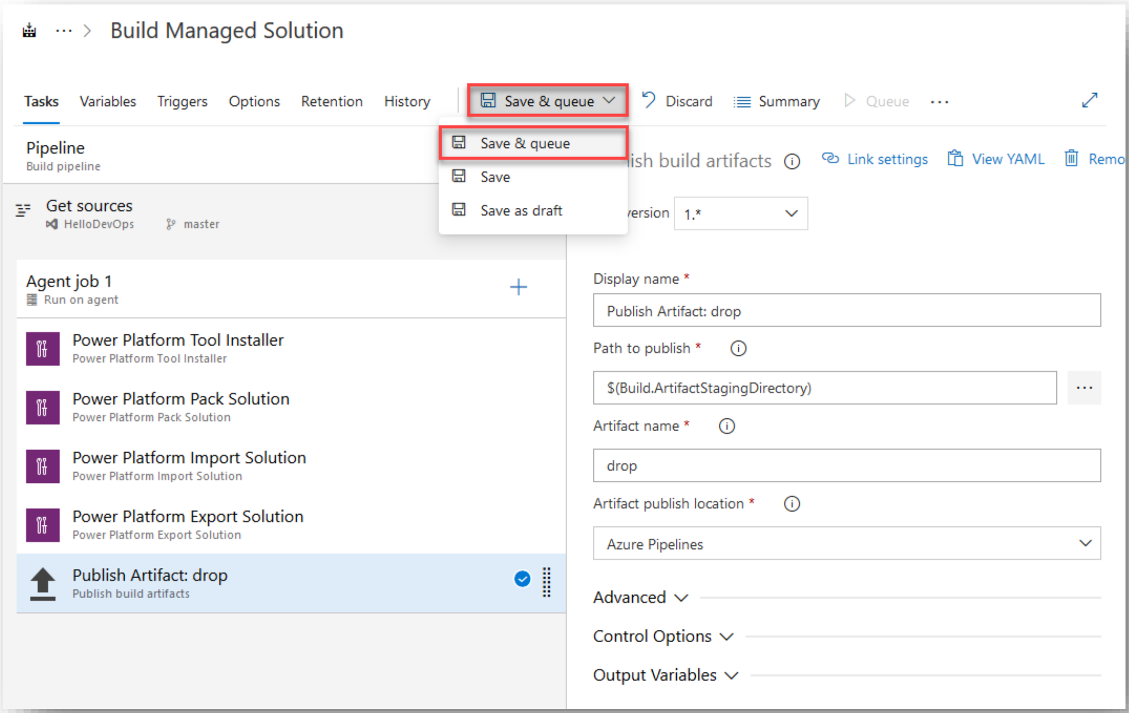
Service connection * | Manage: Build

Solution Name *: $(SolutionName)

Solution Output File *: $(Build.ArtifactStagingDirectory)\$(SolutionName)_managed.zip

- [✓] Export as Managed Solution

72. Leave the predefined values as-is and click Save & Queue:



Version: 1.*

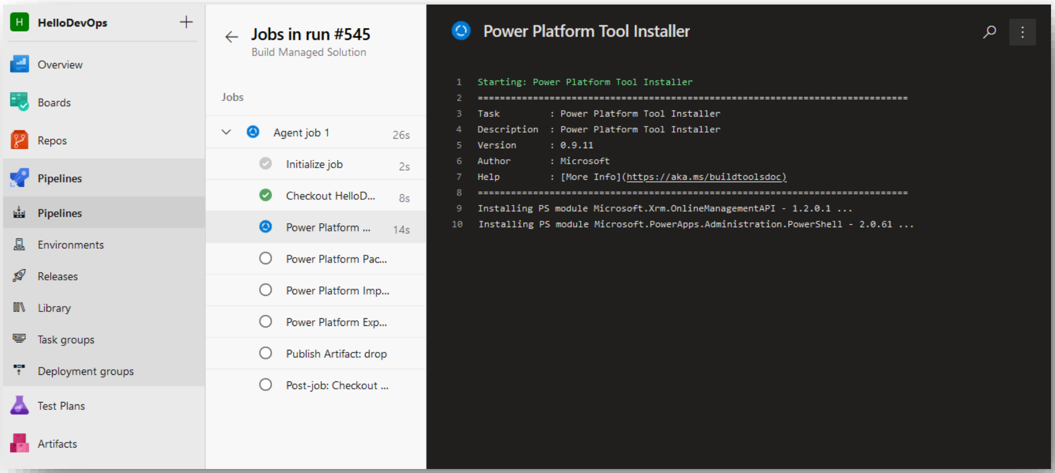
Display name *: Publish Artifact: drop

Path to publish *: $(Build.ArtifactStagingDirectory)

Artifact name *: drop

Artifact publish location *: Azure Pipelines

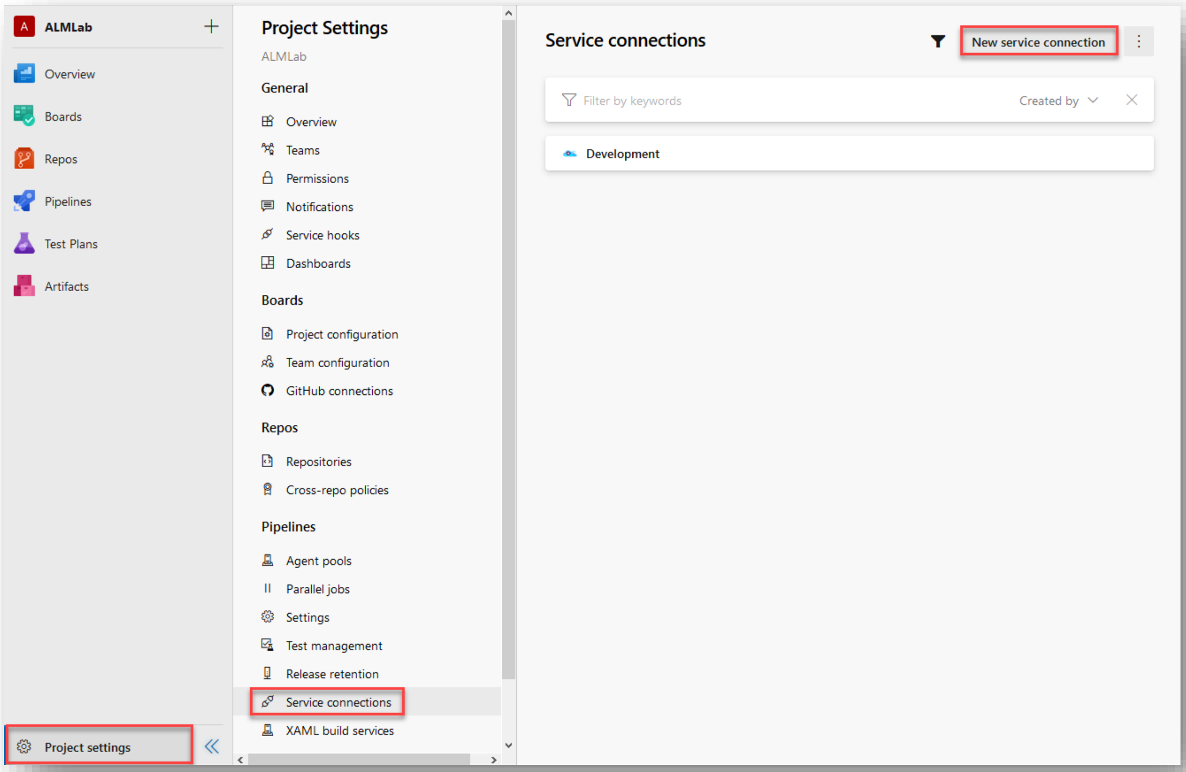
73. Monitor it until it is successful.



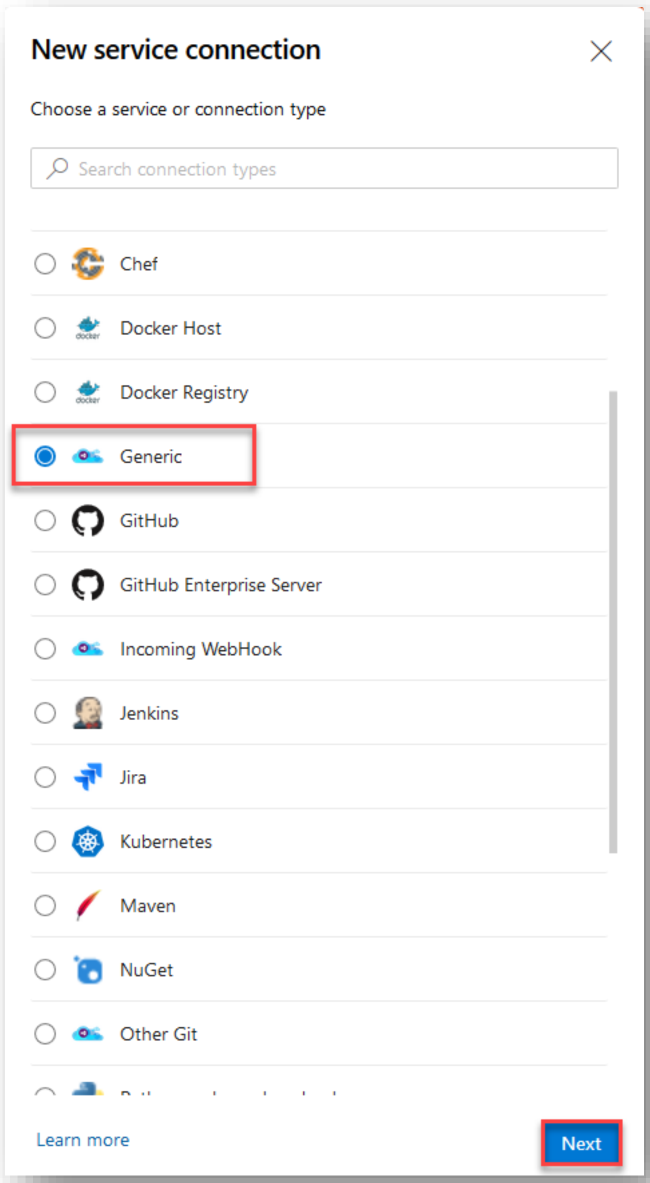
```
Starting: Power Platform Tool Installer
==============================================================================
Task         : Power Platform Tool Installer
Description  : Power Platform Tool Installer
Version      : 0.9.11
Author       : Microsoft
Help         : [More Info](https://aka.ms/buildtoolsdoc)
==============================================================================
Installing PS module Microsoft.Xrm.OnlineManagementAPI - 1.2.0.1 ...
Installing PS module Microsoft.PowerApps.Administration.PowerShell - 2.0.61 ...
```

Release Pipeline: Release to Production

74. The next task will be to import the solution into your production server, so we need to add a connection to that environment before we add and configure the task to do the import. Click the 'Project settings' link in the lower left of the screen and navigate to 'Service connections', then click 'New service connection'.



75. Select 'Generic' on the New service connection page and click 'Next'.


New service connection

Choose a service or connection type

Search connection types

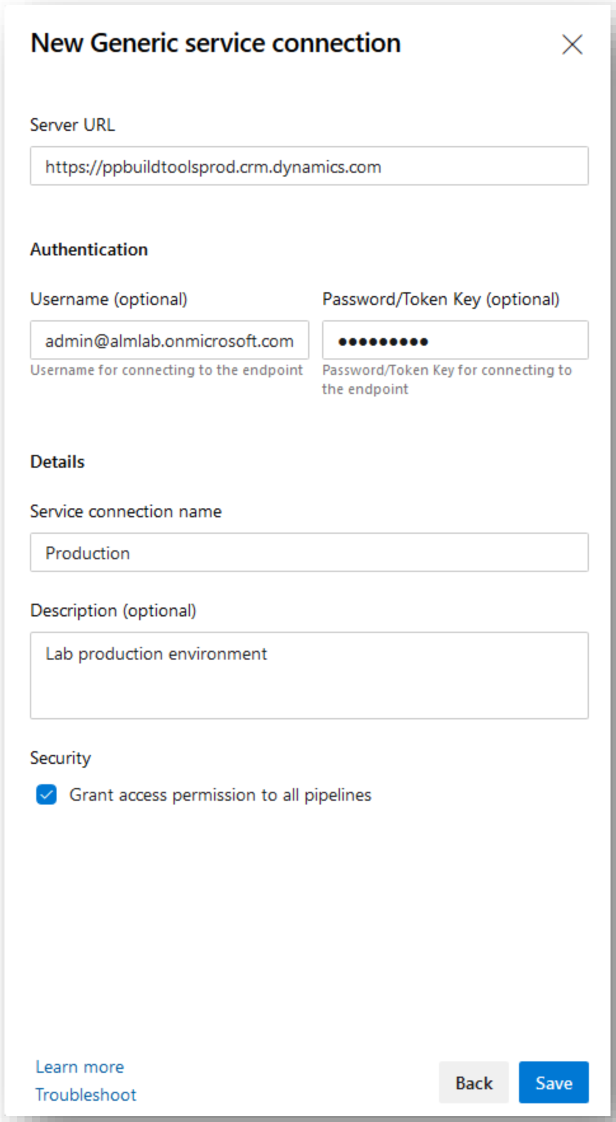
- Chef
- Docker Host
- Docker Registry
- [✓ Generic]
- GitHub
- GitHub Enterprise Server
- Incoming WebHook
- Jenkins
- Jira
- Kubernetes
- Maven
- NuGet
- Other Git

Learn more

Next

76. Fill out the required details:
    - **Server URL**: https://<environment-url>.crm.dynamics.com. This should be the URL for your production environment.
    - **Username:** Username of a user with administrator access to the environment
    - **Password:** Associated password for the user
    - **Service Connection name:** This name will be used to identify which environment to connect to in the Build tasks
    - **Description:** Give the connection a description that helps you identify the environment the service connection connects to

    The sample connection is shown below.



New Generic service connection

Server URL: https://ppbuildtoolsprod.crm.dynamics.com

Authentication

Username (optional): admin@almlab.onmicrosoft.com
Username for connecting to the endpoint

Password/Token Key (optional): ●●●●●●●●●●
Password/Token Key for connecting to the endpoint

Details

Service connection name: Production

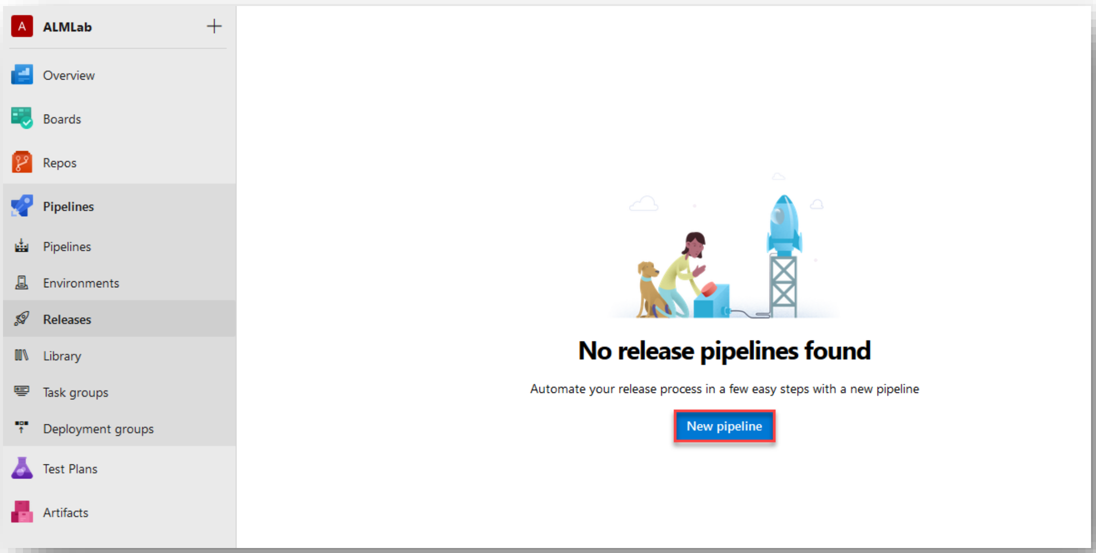
Description (optional): Lab production environment

Security

✓ Grant access permission to all pipelines

77. Click 'Save'. You will be in the pipeline service connections area in your project.

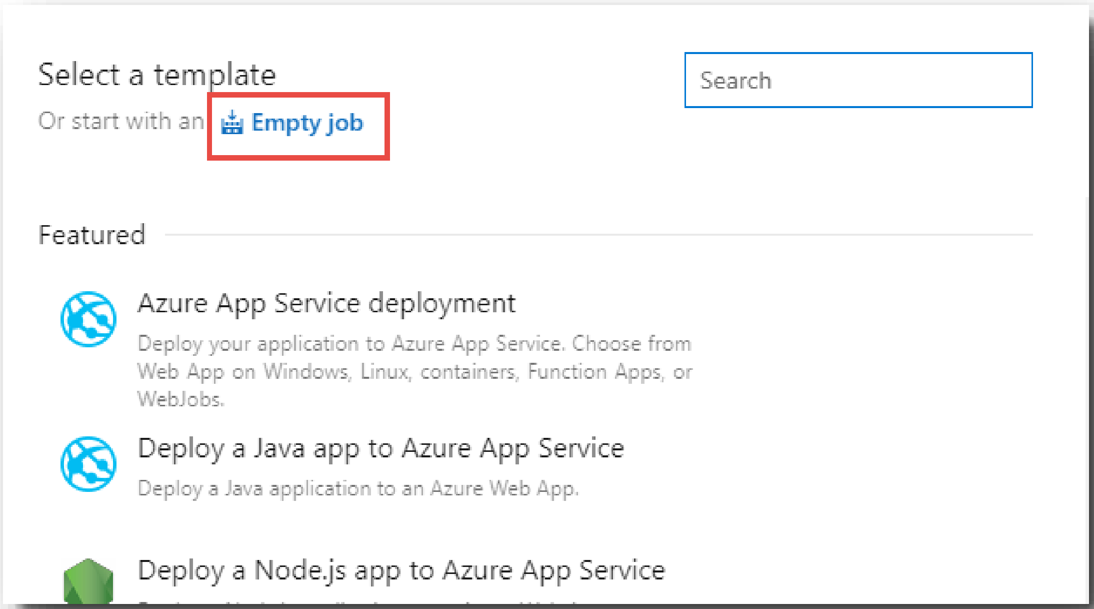
78. To deploy a build, you will configure a release pipeline. Navigate to the release pipelines. You likely won't have one yet, so click on the New pipeline button.



No release pipelines found

Automate your release process in a few easy steps with a new pipeline

79. Click on Empty job when selecting a template.



Azure App Service deployment

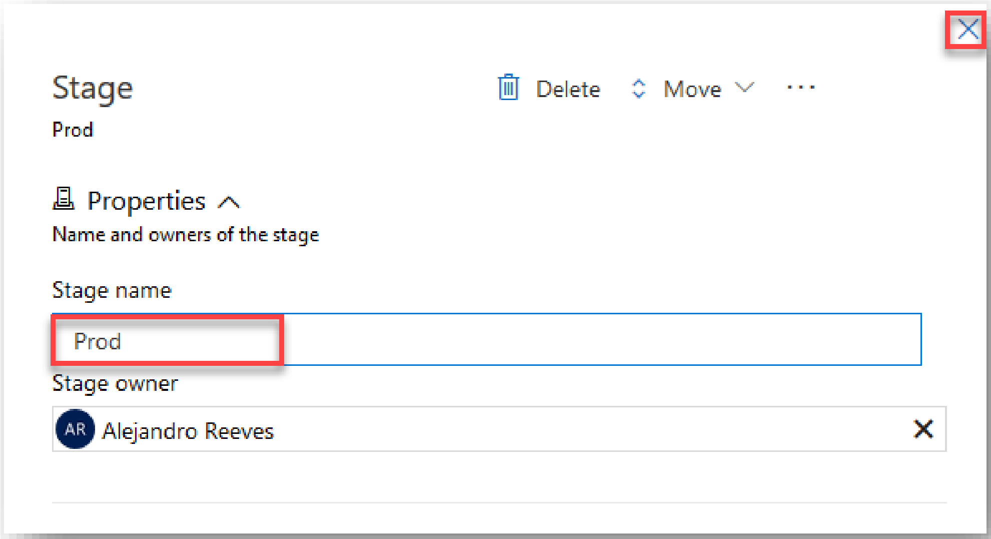
Deploy your application to Azure App Service. Choose from Web App on Windows, Linux, containers, Function Apps, or WebJobs.

Deploy a Java app to Azure App Service

Deploy a Java application to an Azure Web App.

Deploy a Node.js app to Azure App Service

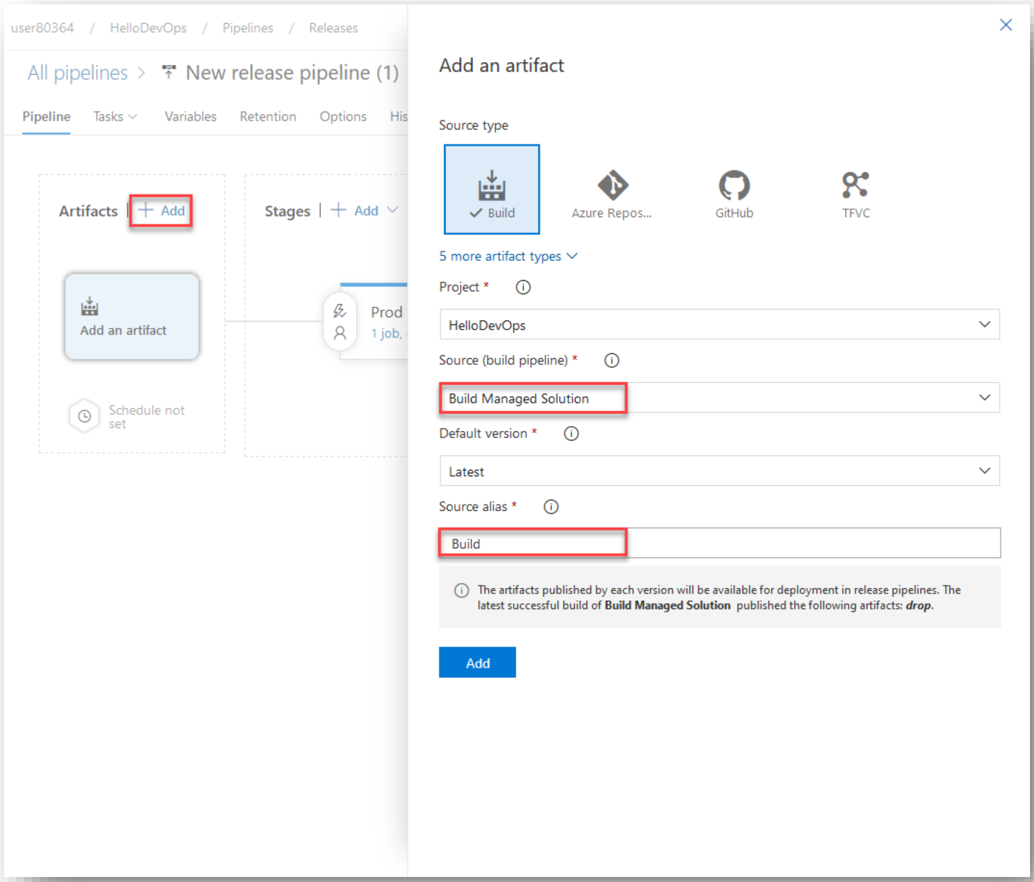
80. Give it a stage name and then close the stage dialogue.



Stage name: Prod

Stage owner: Alejandro Reeves

81. Next, we need to add the artifact that will be used in the deployment. Click the Add button in the Artifacts list. In the Add an artifact screen, select your 'Build Managed Solution' from the source build pipeline. To make it easier to not have to deal with encoding spaces in filenames, change your source alias to something simple, like Build. Leave the rest as default. Click Add when completed.



Add an artifact

Project * HelloDevOps

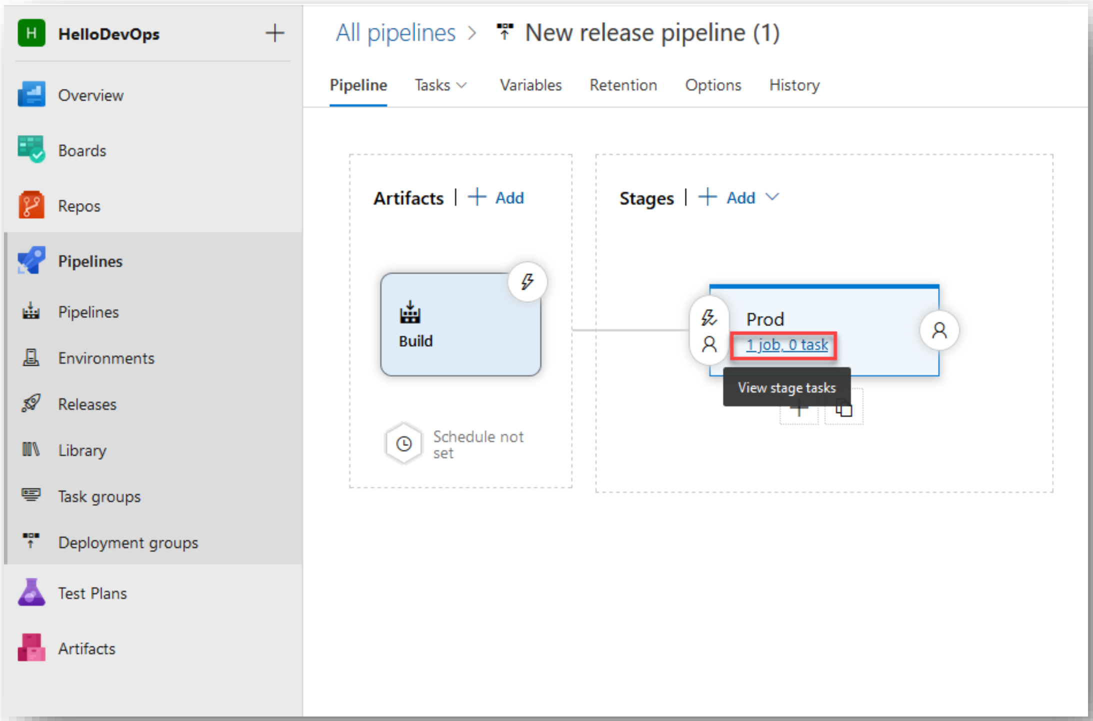
Source (build pipeline) * Build Managed Solution

Default version * Latest

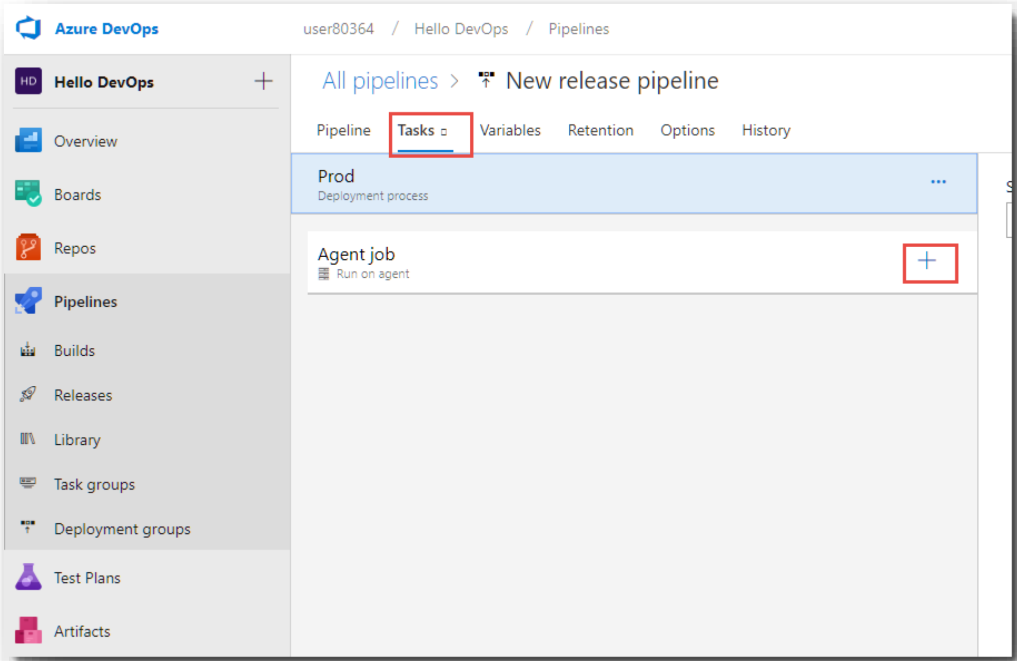
Source alias * Build

The artifacts published by each version will be available for deployment in release pipelines. The latest successful build of **Build Managed Solution** published the following artifacts: ***drop***.

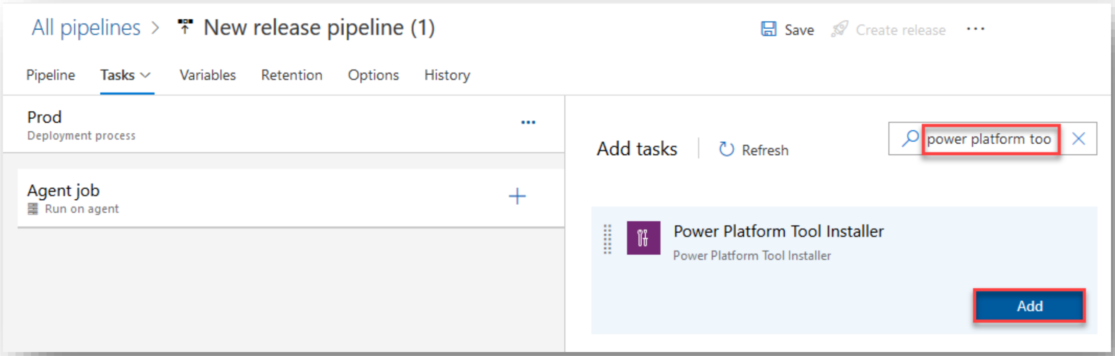
82. Switch to the Tasks view and click the plus button to add a new task.



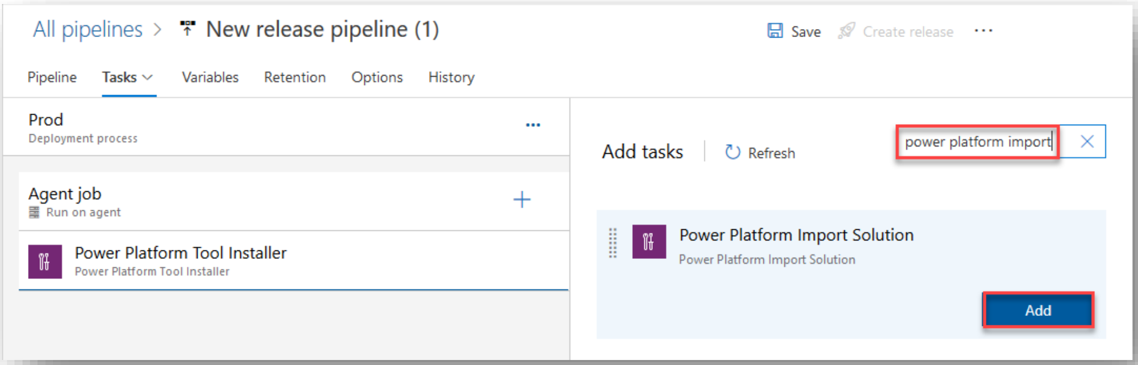
83. In the tasks view, click the plus button to add a new task.



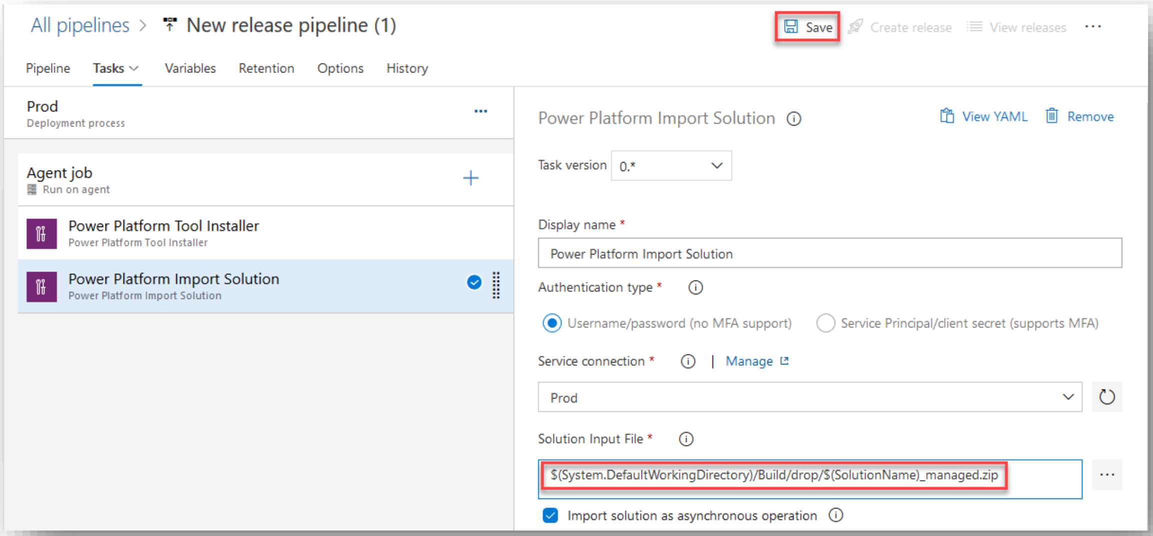
84. Add the Power Platform Tool Installer tasks.



85. Add Power Platform Import Solution task to the pipeline.

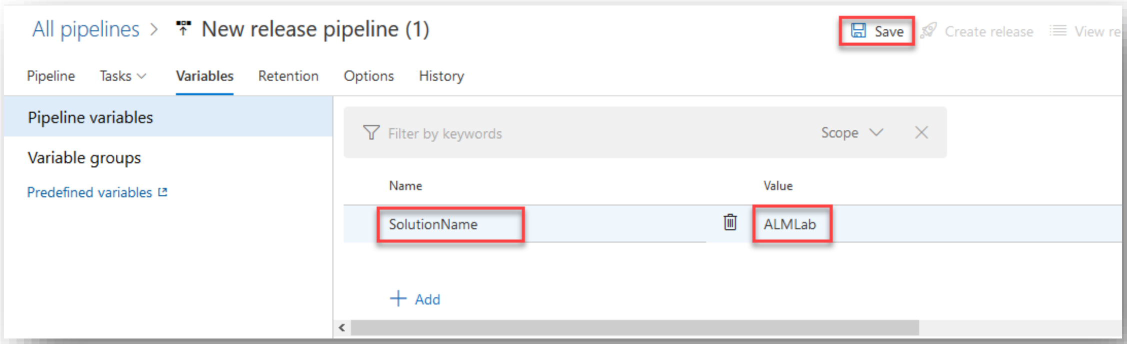

All pipelines > New release pipeline (1)
Save
Create release
Pipeline
Tasks
Variables
Retention
Options
History
Prod
Deployment process
Agent job
Run on agent
Power Platform Tool Installer
Power Platform Tool Installer
Add tasks
Refresh
power platform import
Power Platform Import Solution
Power Platform Import Solution
Add

86. Configure the solution import to use the production environment connection and the following for solution input file:
$(System.DefaultWorkingDirectory)/Build/drop/$(SolutionName)_managed.zip
Click ‘Save’.


All pipelines > New release pipeline (1)
Save
Create release
View releases
Pipeline
Tasks
Variables
Retention
Options
History
Prod
Deployment process
Agent job
Run on agent
Power Platform Tool Installer
Power Platform Tool Installer
Power Platform Import Solution
Power Platform Import Solution
Power Platform Import Solution
View YAML
Remove

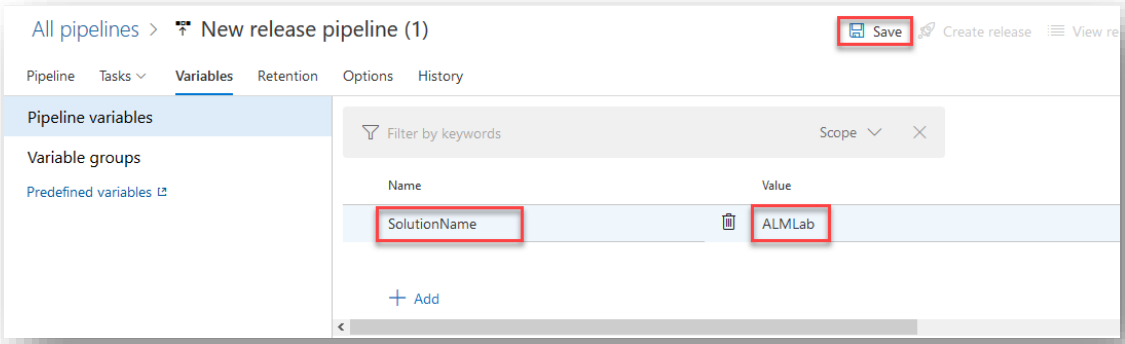
Task version: 0.*
Display name *: Power Platform Import Solution
Authentication type *: [✓ Username/password (no MFA support)] Service Principal/client secret (supports MFA)
Service connection * | Manage: Prod
Solution Input File *: $(System.DefaultWorkingDirectory)/Build/drop/$(SolutionName)_managed.zip
[✓] Import solution as asynchronous operation

87. Switch to the Variables tab and add a variable named SolutionName and the unique name of the solution to import. If you are using the solution created in lab 1, the solution name should be 'ALMLab'. Click 'Save' and then 'OK' in the next dialog.



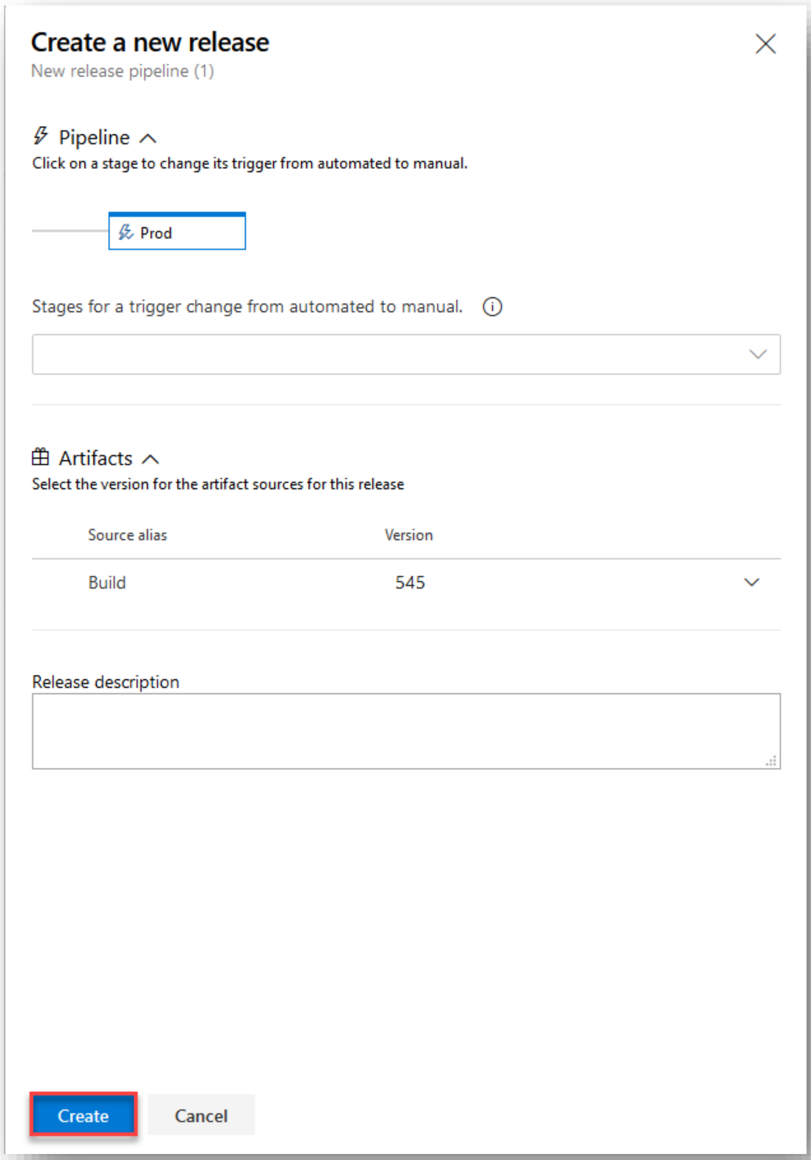
Name	Value
SolutionName	ALMLab

88. Click the "Create release" button. This functions like Queuing a build pipeline, only for release pipelines.



Name	Value
SolutionName	ALMLab

89. Click ‘Create’ on the popup panel.



Create a new release
New release pipeline (1)

Pipeline
Click on a stage to change its trigger from automated to manual.

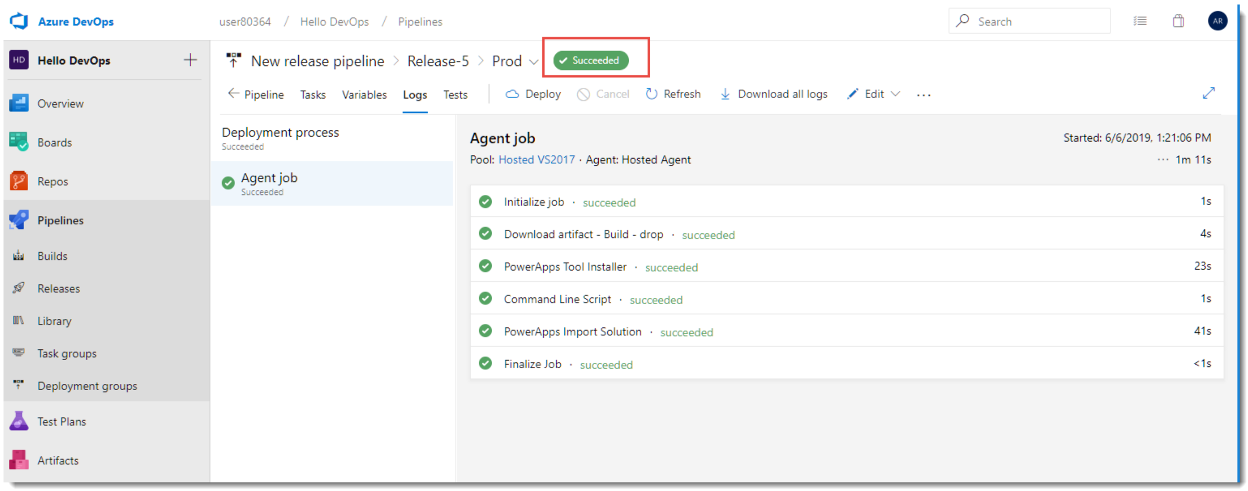
Stages for a trigger change from automated to manual.

Artifacts
Select the version for the artifact sources for this release

Source alias	Version
Build	545

Release description

90. Monitor your release until it is succeeded, or troubleshoot if it is not successful.



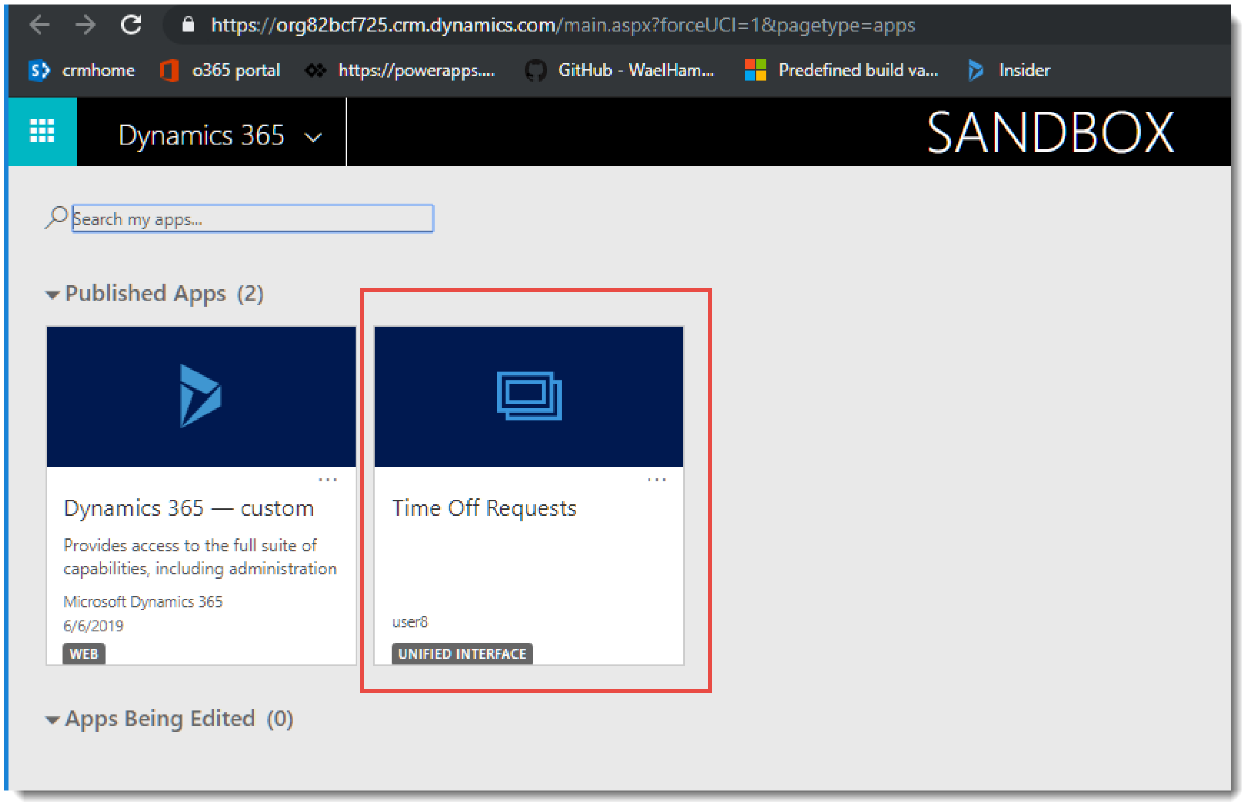
Deployment process
Succeeded

Agent job
Succeeded

Agent job
Pool: Hosted VS2017 · Agent: Hosted Agent
Started: 6/6/2019, 1:21:06 PM
··· 1m 11s

Initialize job · succeeded	1s
Download artifact - Build - drop · succeeded	4s
PowerApps Tool Installer · succeeded	23s
Command Line Script · succeeded	1s
PowerApps Import Solution · succeeded	41s
Finalize Job · succeeded	<1s

91. For final confirmation, log into your production system and see your application!



Published Apps (2)

Dynamics 365 — custom
Provides access to the full suite of capabilities, including administration
Microsoft Dynamics 365
6/6/2019
WEB

Time Off Requests
user8
UNIFIED INTERFACE

Apps Being Edited (0)

Terms of Use

FEEDBACK

DISCLAIMER