Lab 1: Setup and Configure

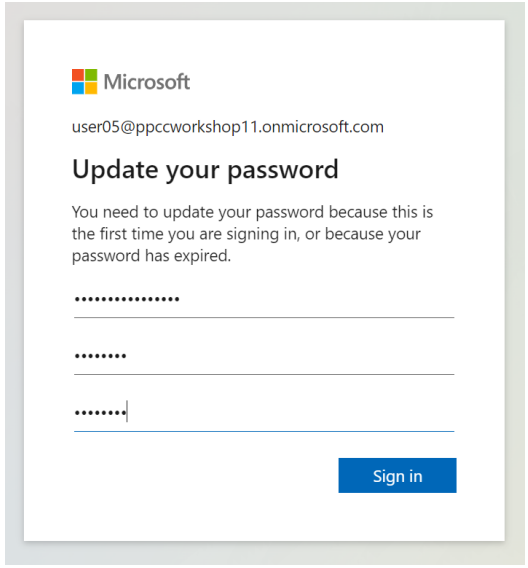
Lab 1 – Tasks

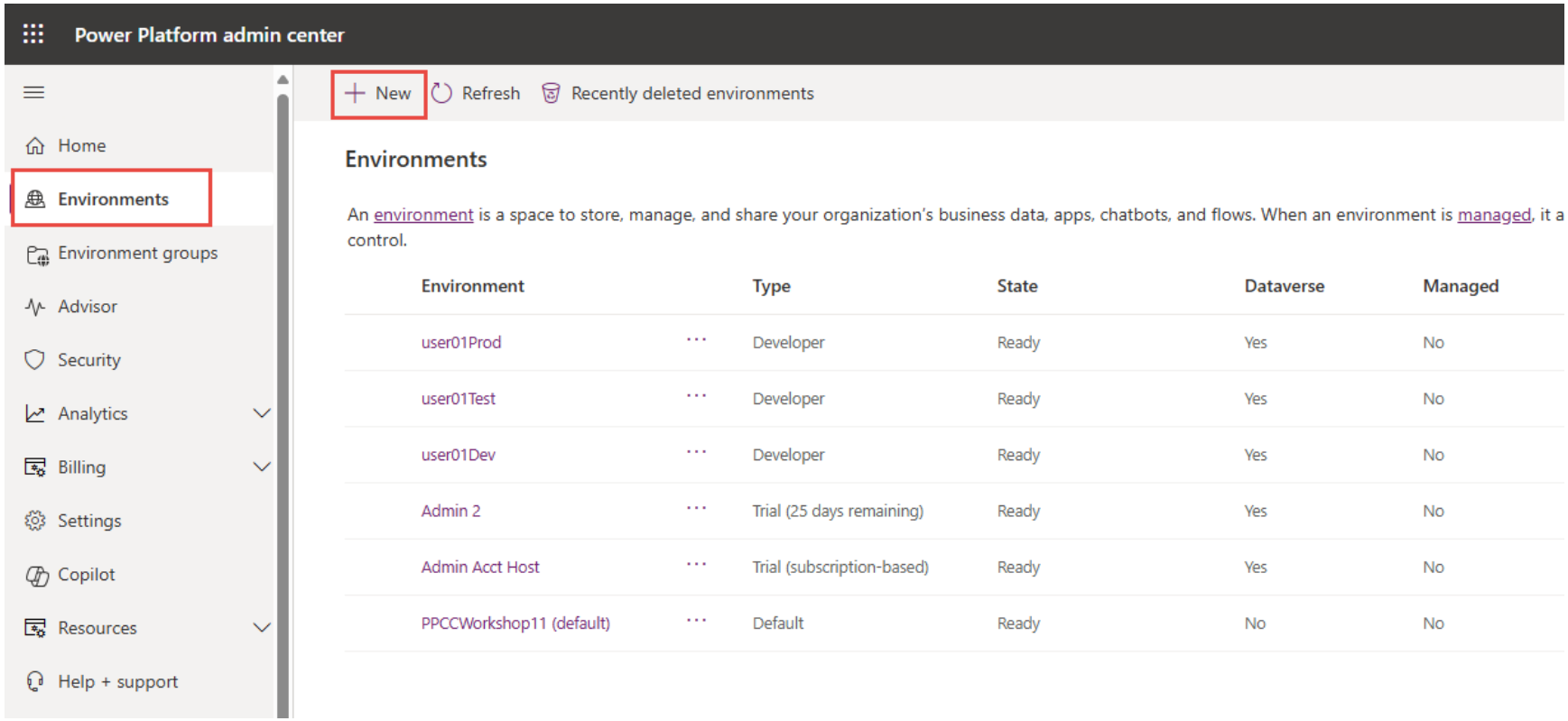
- Create environments
- Configure Azure DevOps repository
- Enable Native Git Integration

Task 1 - Create Environments

Create three environments

1. Navigate to https://admin.powerplatform.microsoft.com/environments
2. Login using the workshop account on your desk
3. When prompted, change your password


Microsoft
user05@ppccworkshop11.onmicrosoft.com
Update your password
You need to update your password because this is the first time you are signing in, or because your password has expired.
Sign in



2. Click **+New** to open the environment creation side panel.

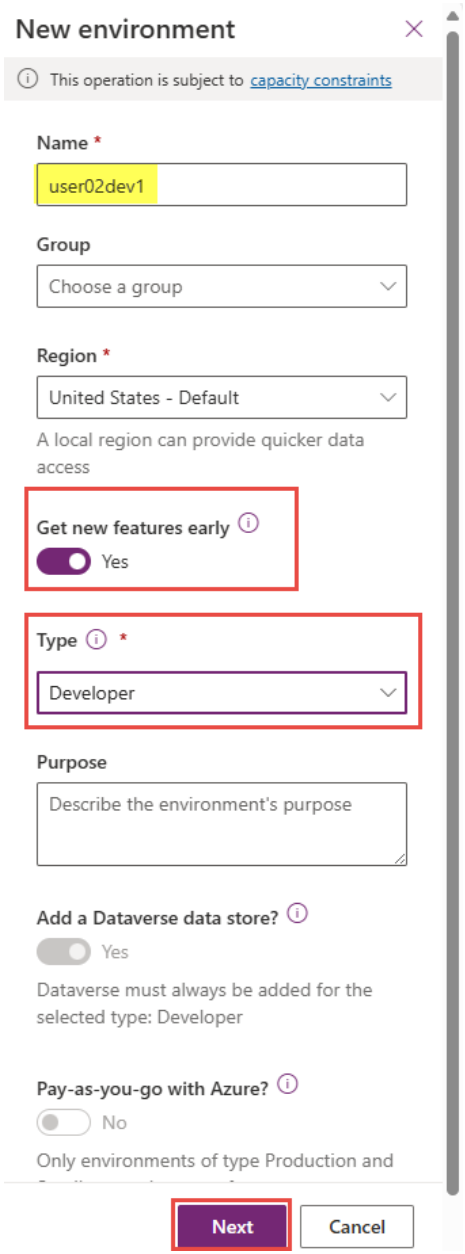
Give the environment a name (suggested *userXXdev1* where userXX corresponds to your username provided for the workshop).

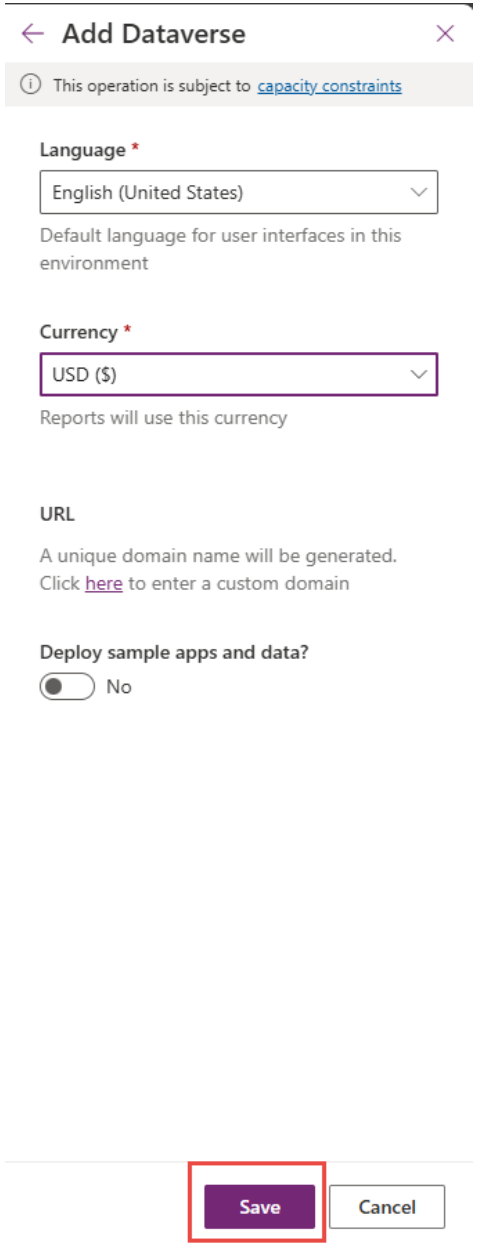
Enable the **Get new features early** option. This is ***CRITICAL*** for this lab as the source code and other features are limited to the early release cycle.

Change the environment type to **Developer.** This will also ensure that you have a Dataverse database provided as well.

Note: For the purpose of this lab it is important for you to pick this type as you need to leverage the license ability to create three personal environments for your user as the tenant does not have the capacity to provision a number of sandbox or production environments.

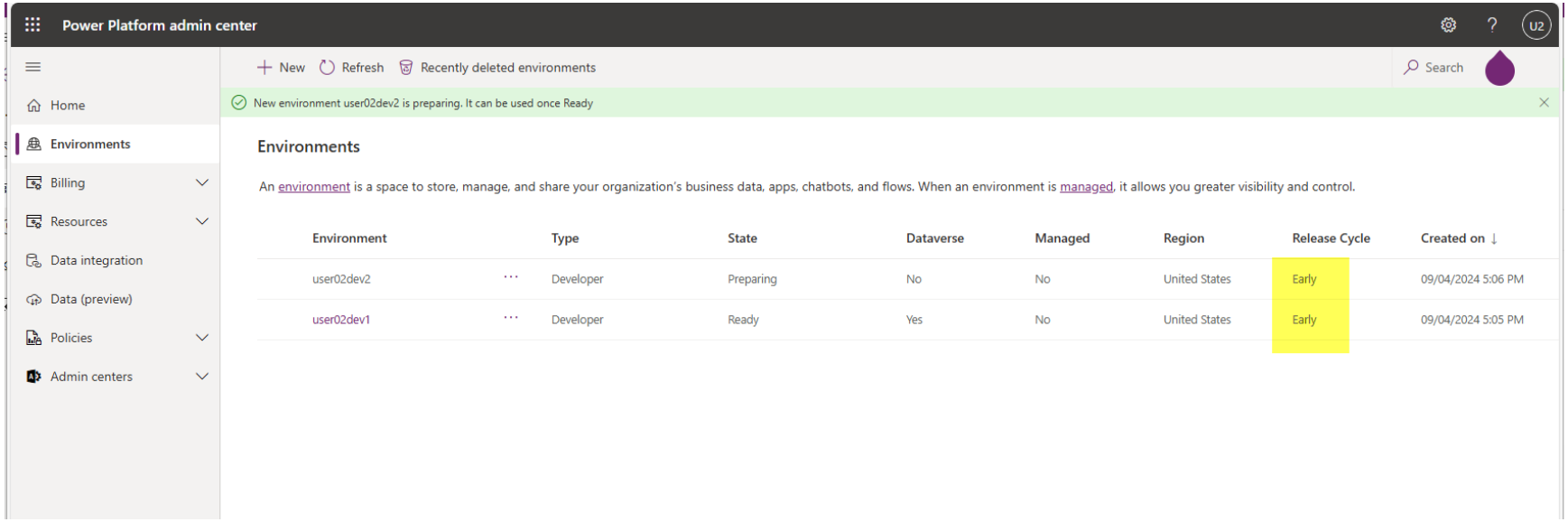
3. Complete the environment creation by clicking **Next** then on the next panel, click **Save**.


New environment
This operation is subject to capacity constraints
Name *
user02dev1
Group
Choose a group
Region *
United States - Default
A local region can provide quicker data access
Get new features early
Yes
Type *
Developer
Purpose
Describe the environment's purpose
Add a Dataverse data store?
Yes
Dataverse must always be added for the selected type: Developer
Pay-as-you-go with Azure?
No
Only environments of type Production and
Next
Cancel


Add Dataverse
This operation is subject to capacity constraints
Language *
English (United States)
Default language for user interfaces in this environment
Currency *
USD ($)
Reports will use this currency
URL
A unique domain name will be generated.
Click here to enter a custom domain
Deploy sample apps and data?
No
Save
Cancel

4. Create a second development environment using the above process. For the second environment, use the exact same settings but give it a unique name such as UserXXdev2.

Verify that both environments were created successfully AND that the Release Cycle is set to Early.

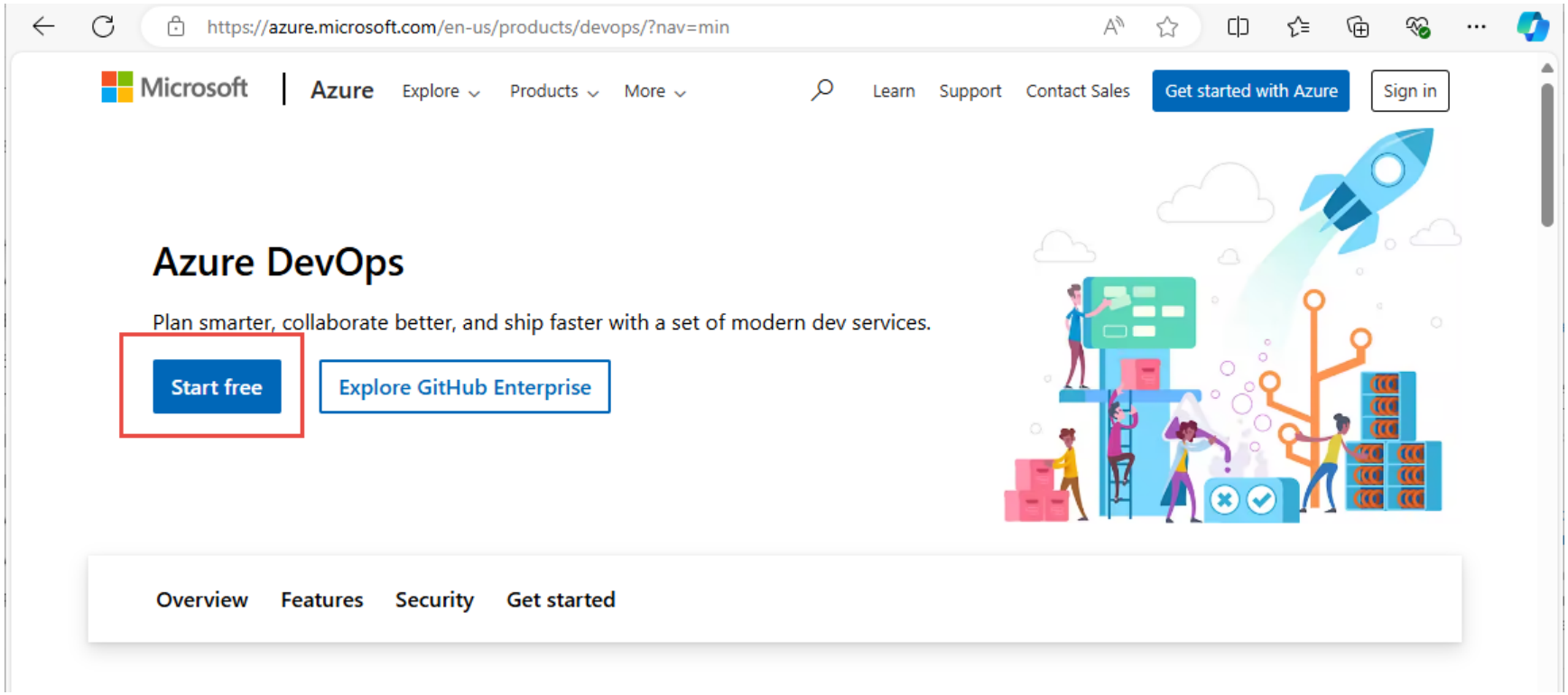


5. Create a third environment using the above process. For the third environment, use the exact same settings but give it a unique name such as UserXXProduction. While this environment must be a developer type due to licenses in this tenant, we'll pretend it's a production environment you'll deploy to in lab 3.

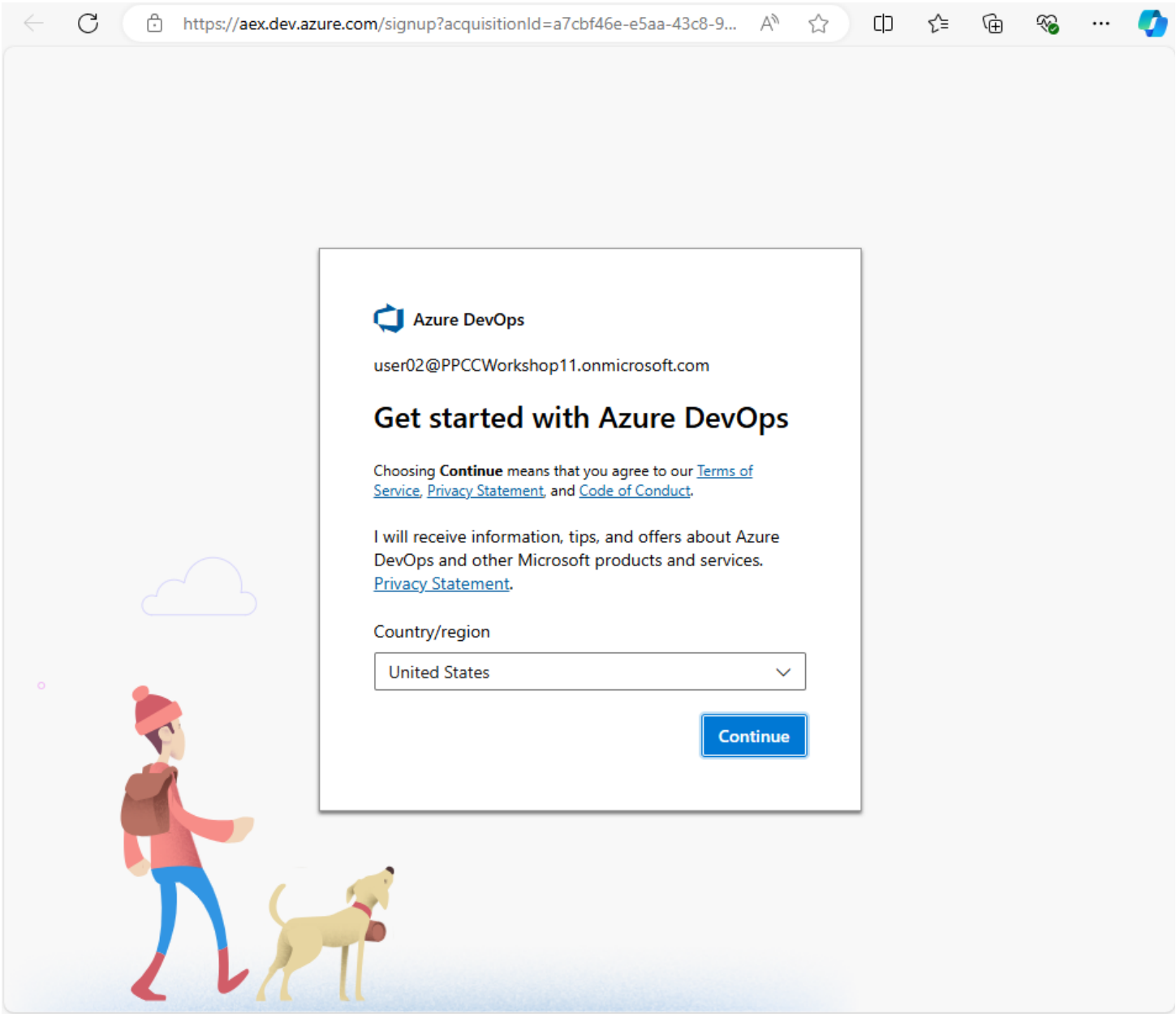
You should now have **three environments**. Two will be used for development and the other for production.

Configure Azure DevOps Repository

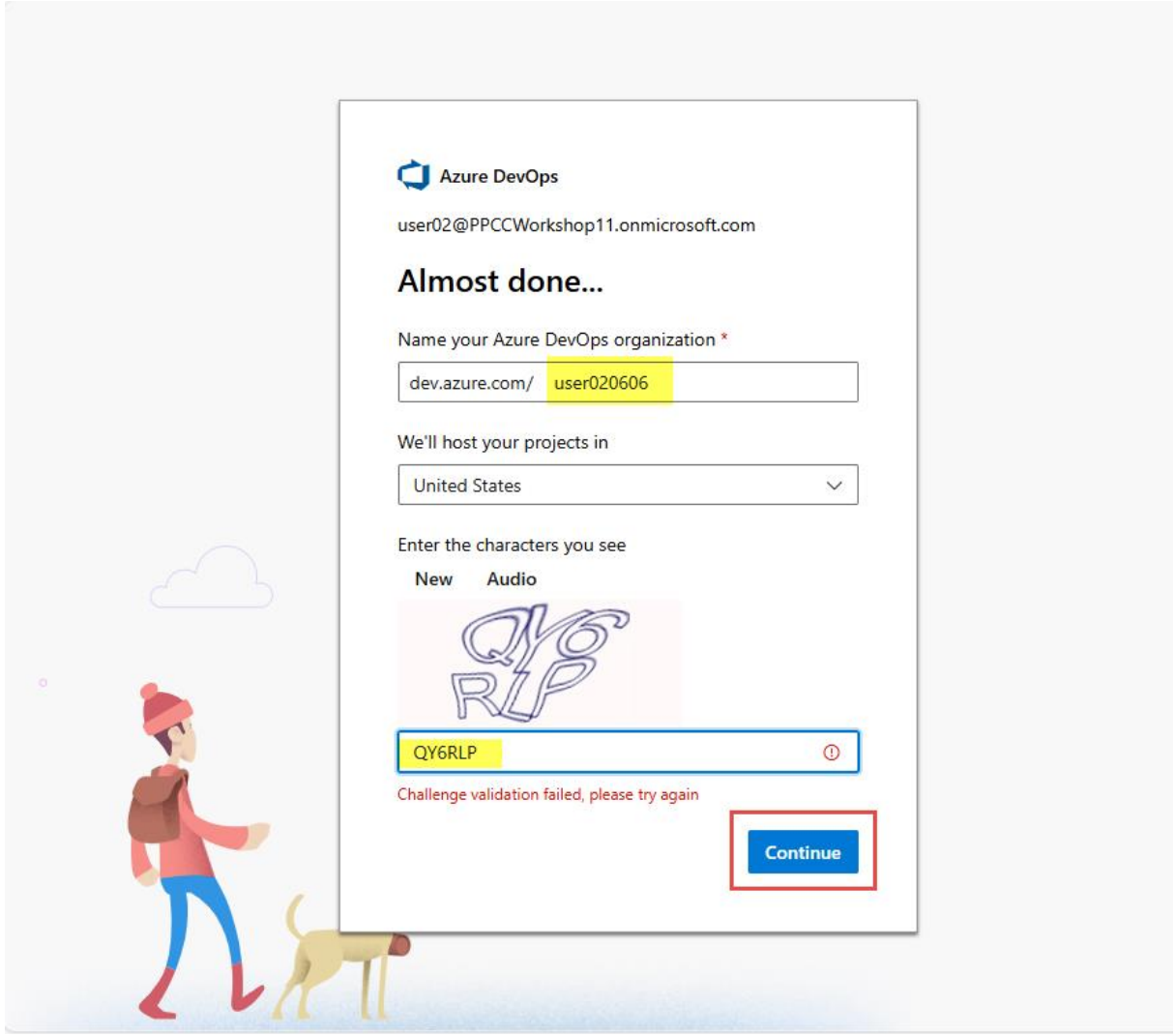
Navigate to https://dev.azure.com/ and click on the Start free button.



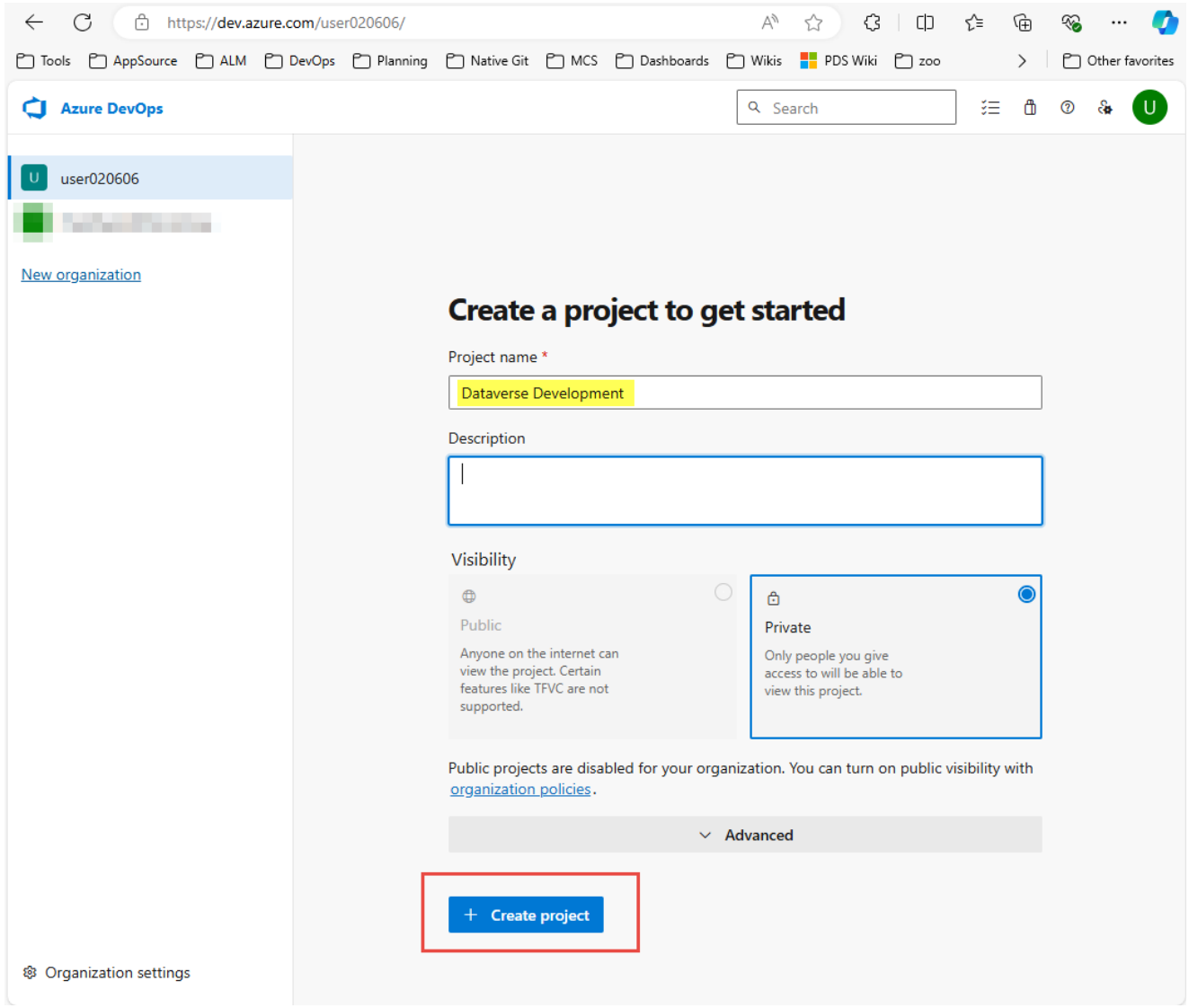
Sign in if you are prompted. The next screen should be your initial DevOps experience. Click Continue.


https://aex.dev.azure.com/signup?acquisitionId=a7cbf46e-e5aa-43c8-9...
Azure DevOps
user02@PPCCWorkshop11.onmicrosoft.com
Get started with Azure DevOps
Choosing Continue means that you agree to our Terms of Service, Privacy Statement, and Code of Conduct.
I will receive information, tips, and offers about Azure DevOps and other Microsoft products and services. Privacy Statement.
Country/region
United States
Continue

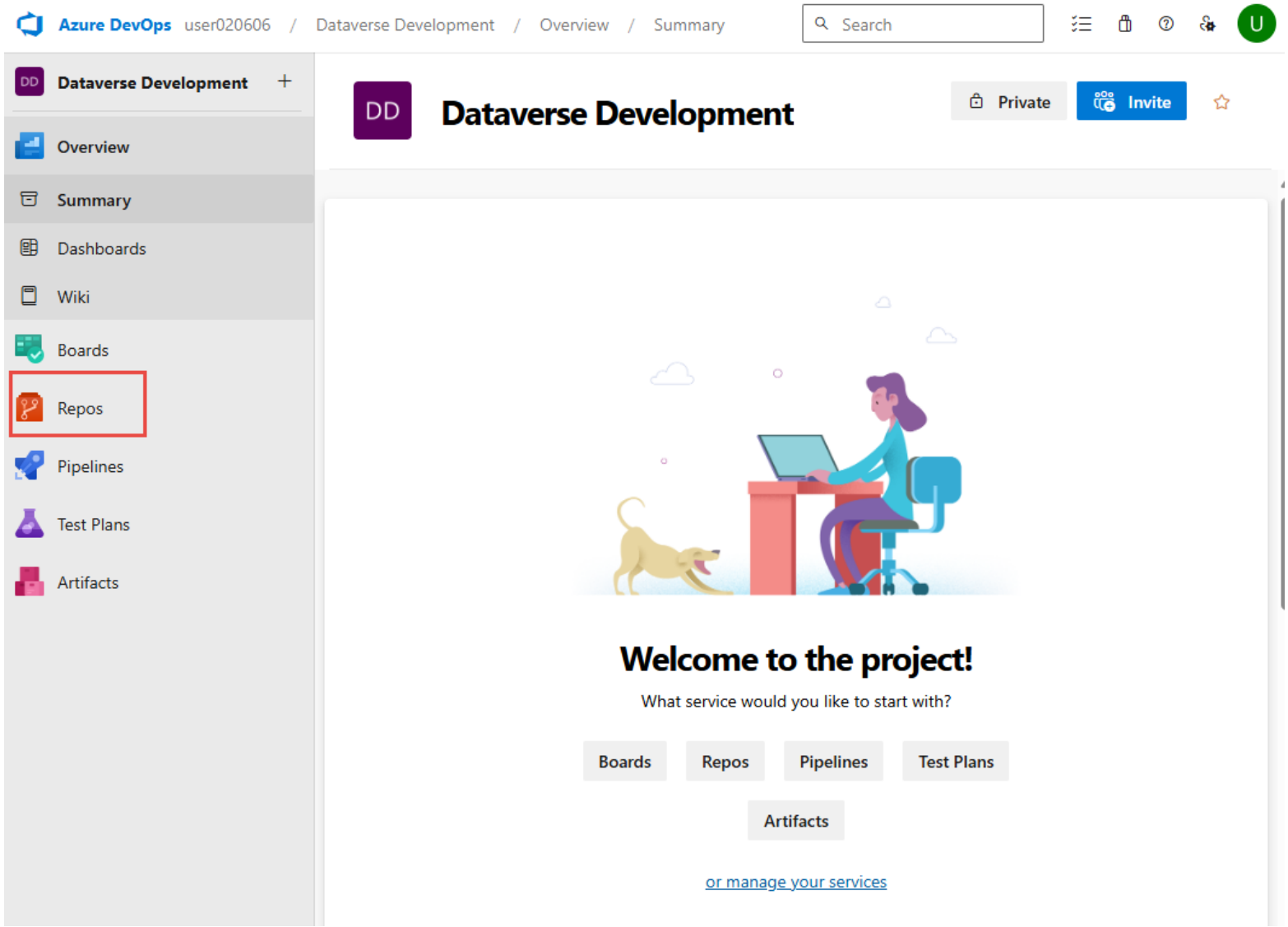
You will have to create an Azure DevOps organization. The experience will suggest a name for you based on your username. Review that and enter the captcha challenge and click the **Continue** button.



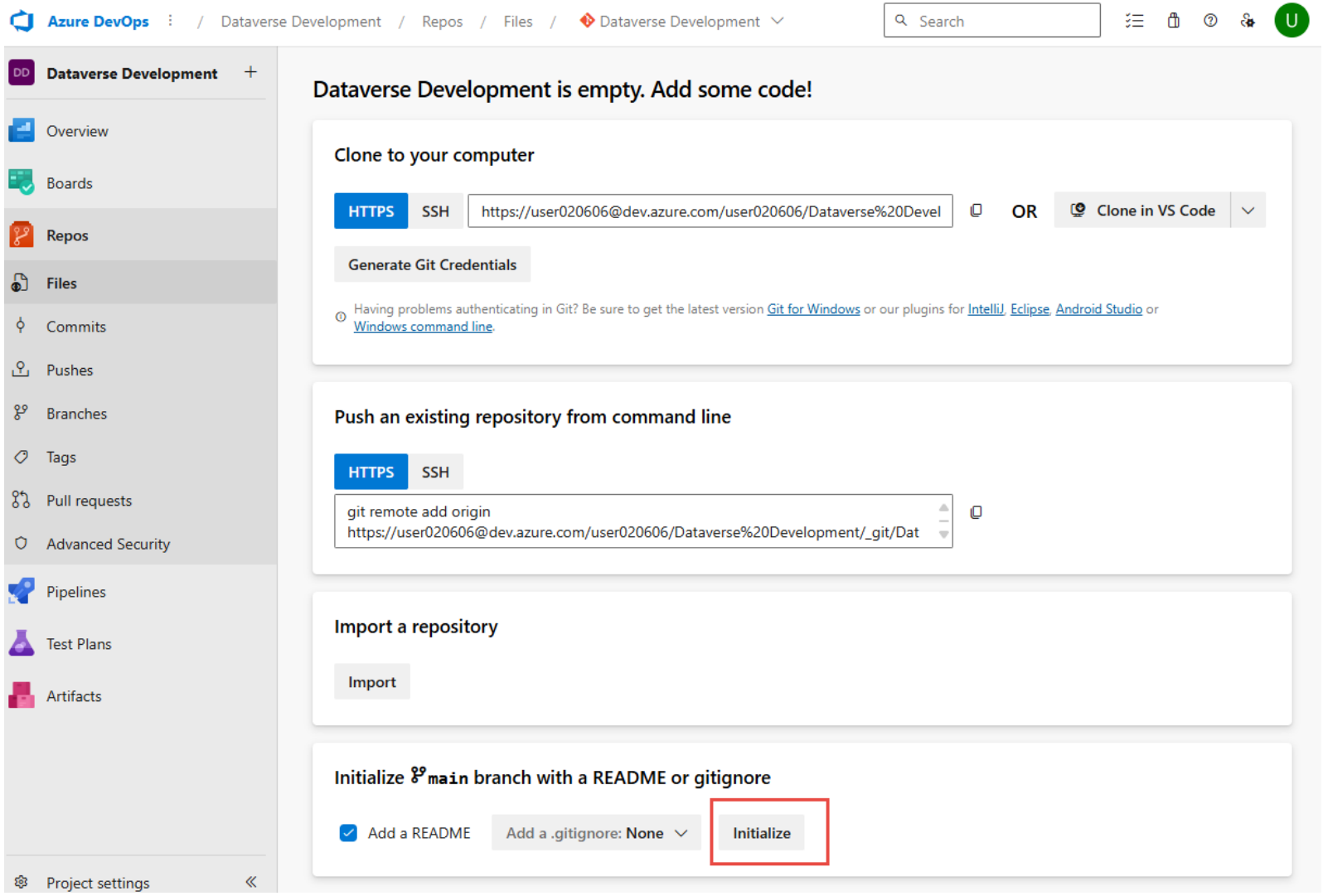
After your organization is created, you will be prompted to create a Azure DevOps project. Enter a project name and click the **Create project** button.



Your next step will be to create and initialize a repository. On the project page, click the **Repos** link.



Then scroll to the bottom of the page and click the **Initialize** button to initialize the repo with a readme file.

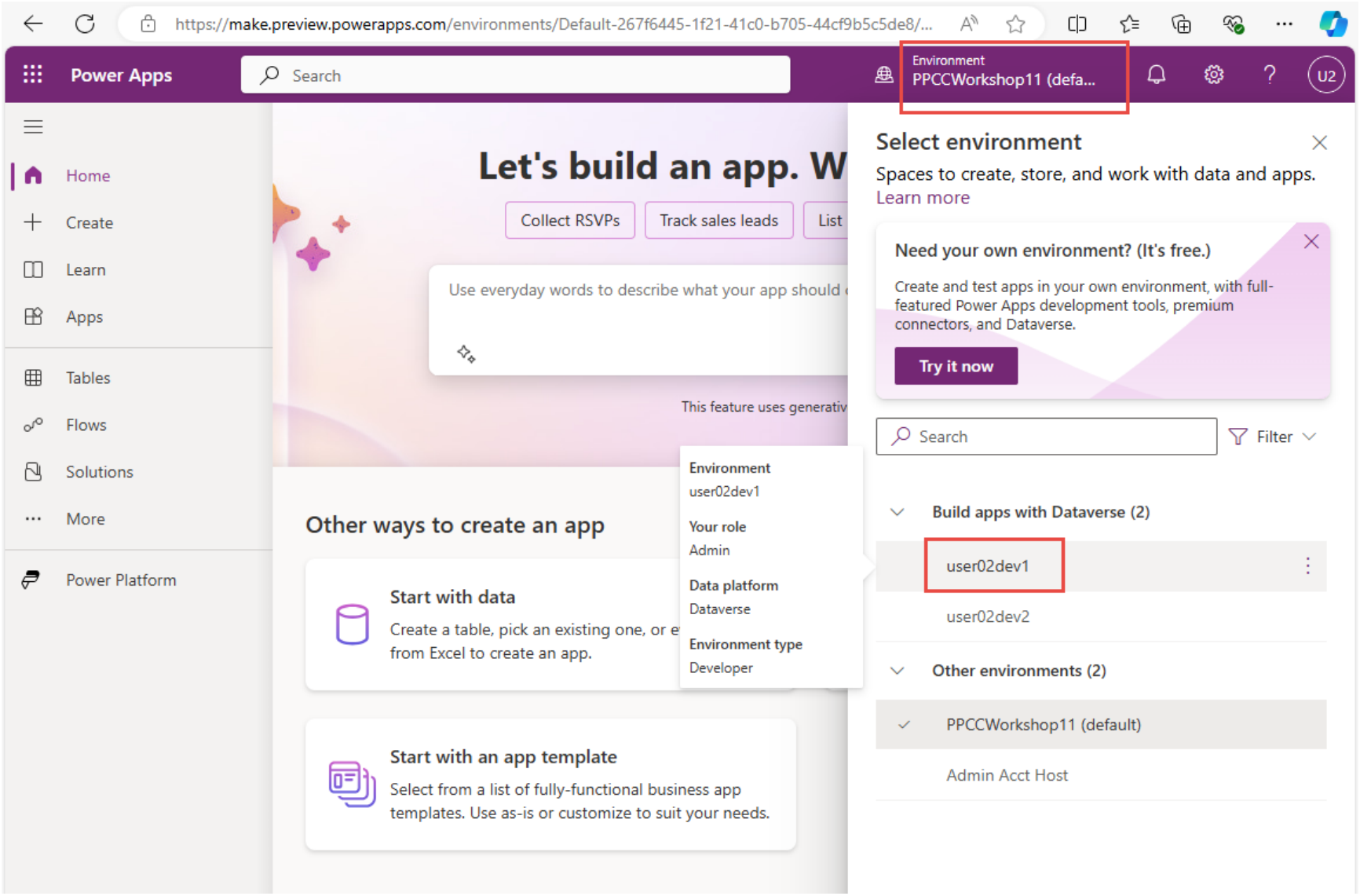


Configure Git Integration on your dev1 environment

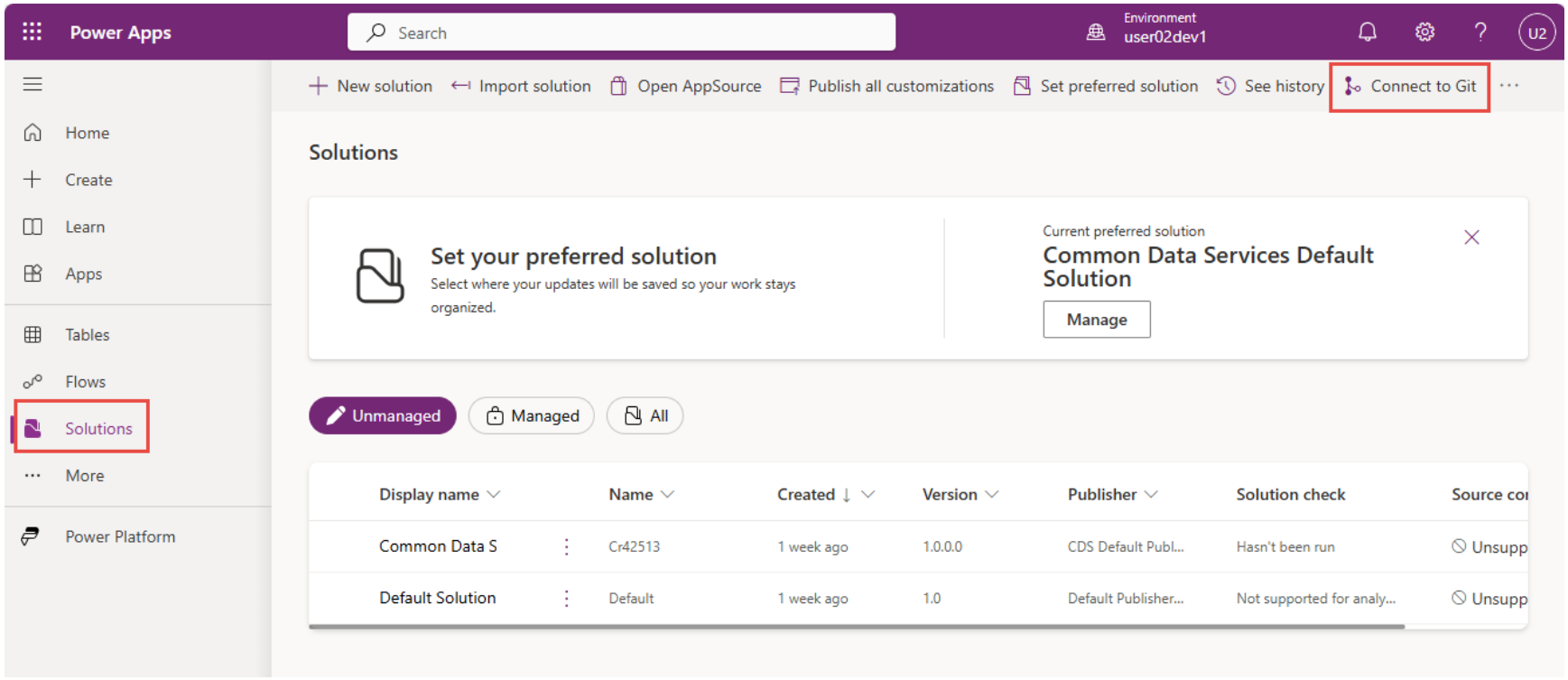
Important! Today you'll be working with Native Git Integration, which is a new capability and **not yet available publicly**. Make sure to use the **below URL**. Otherwise these capabilities will be hidden.

Disclaimer: You shouldn't use preview features in production. This lab is for educational purposes only.

1. Open a new tab and log into the https://make.**preview**.powerapps.com/**?powerappsAlmSourceControl.enableAlmSourceControl=true** using your **workshop credentials**.
2. Use the environment selector and make sure you're in your dev1 environment.



Click on the **Solutions** area in the left navigation bar, then click on the **Connect to Git** button in the command bar.



From here you will connect your Dataverse environment to your Git repository. Select **Environment** as the connection type, and select your Azure DevOps Organization, Project, Repository, and Branch using the dropdown menus. Enter a new folder name for this environment and click the **Connect** button.

Connect to Git

You can connect this environment to a Git repository via Microsoft Azure DevOps.

Solutions developed with source control integration can only be used within Managed Environments. Learn more

Connection type *

- (•) Environment
- ( ) Solution

Connect all solutions in this environment to the same repository, branch and folder in source control.

Organization *

user020606

Select an organization to connect to Git.

Project *

Dataverse Development

Select a project from your selected organization.

Repository *

Dataverse Development

Select a Git repository from your selected project.

Branch *

main

Select a branch from your selected Git repository.

+ Create new branch

Git folder *

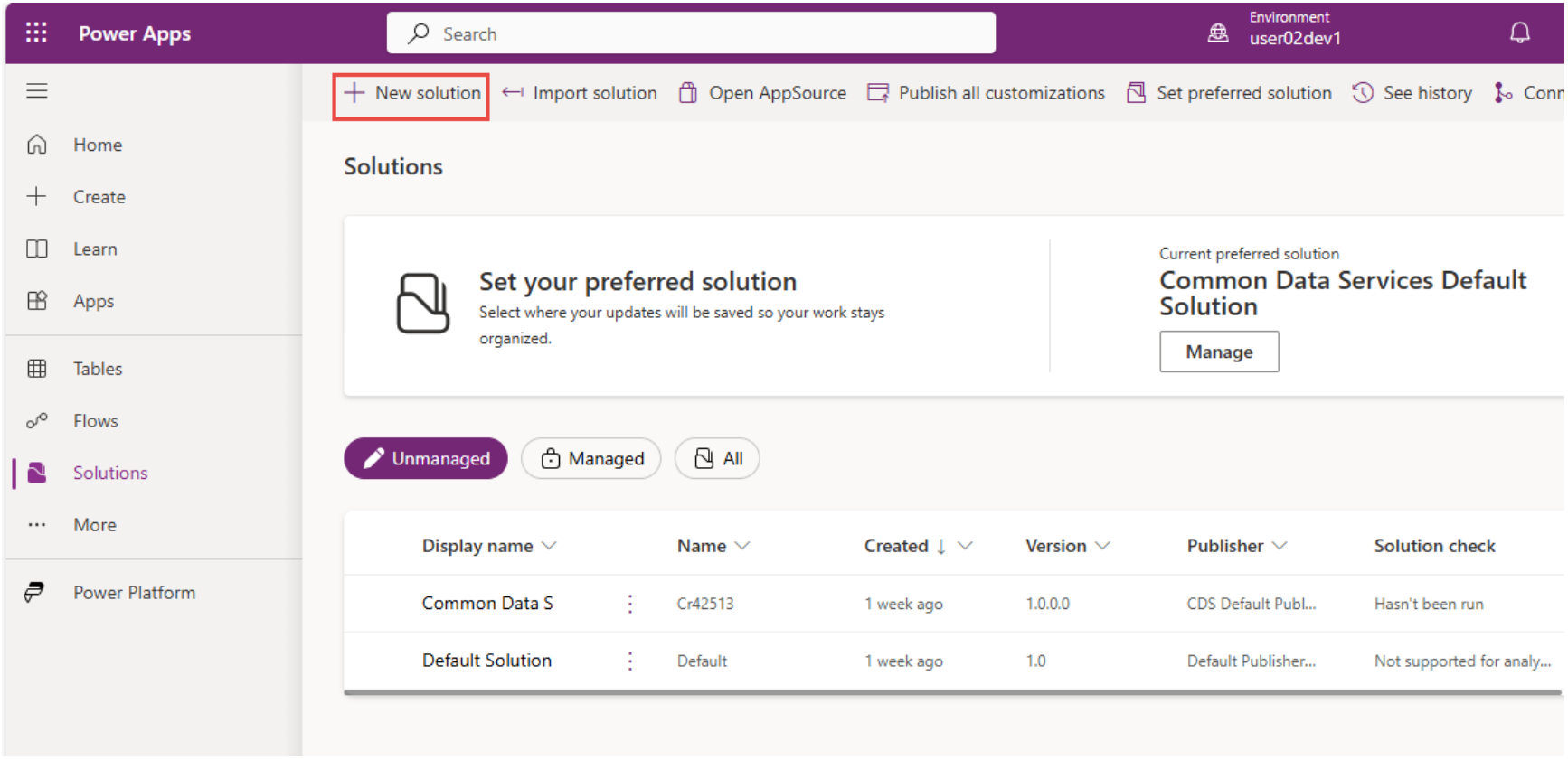
user02

Enter an existing folder from your selected Git repository or create a new one.

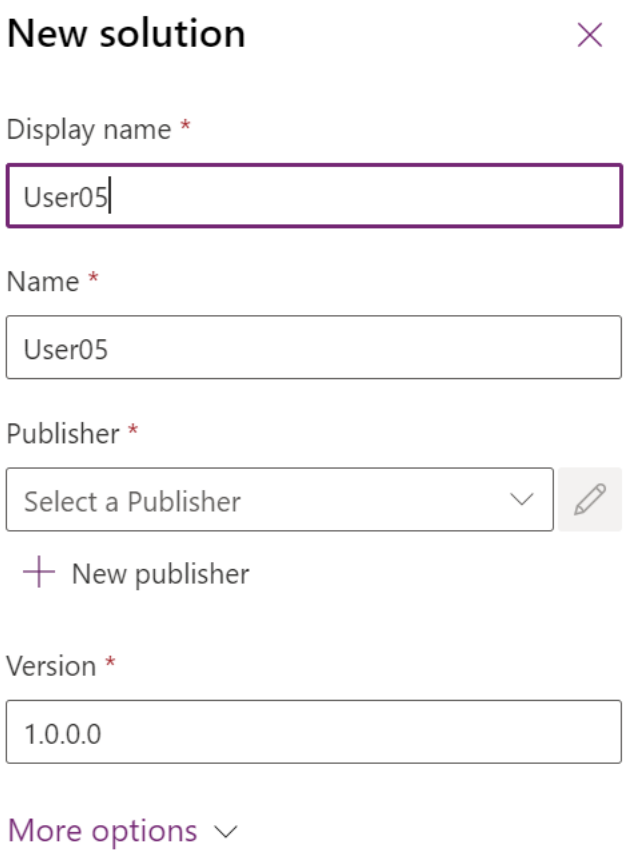
Connect | Cancel

At this point in time, your environment is now bound to Git. Any unmanaged solution changes will now be available to commit to Git. Let's test this by creating an empty solution and doing an initial commit.

Click the **New Solution** button on the command bar.



Use your username in the display name and name.



Click **New publisher** to create a publisher record. Use your workshop user's name for Display name, Name, and Prefix.

Note: many organizations use different naming conventions to identify solutions. We've chosen this format so that it's easy to differentiate your solution from others later.

New publisher

Publishers indicate who developed associated solutions. Learn more

Properties Contact

Display name *

User05

Name *

User05

Description

Prefix *

user05

Choice value prefix *

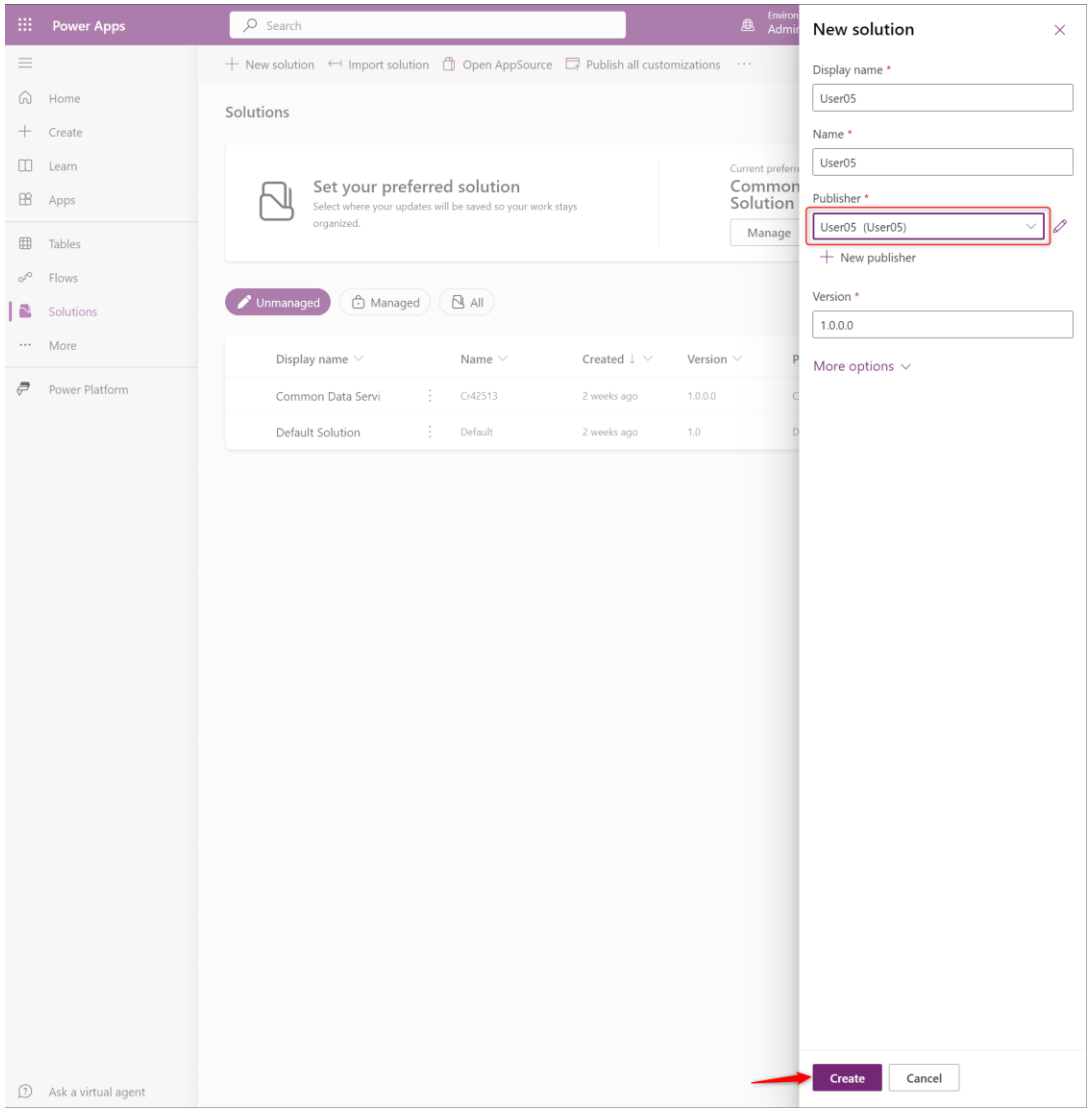
81296

Preview of new object name

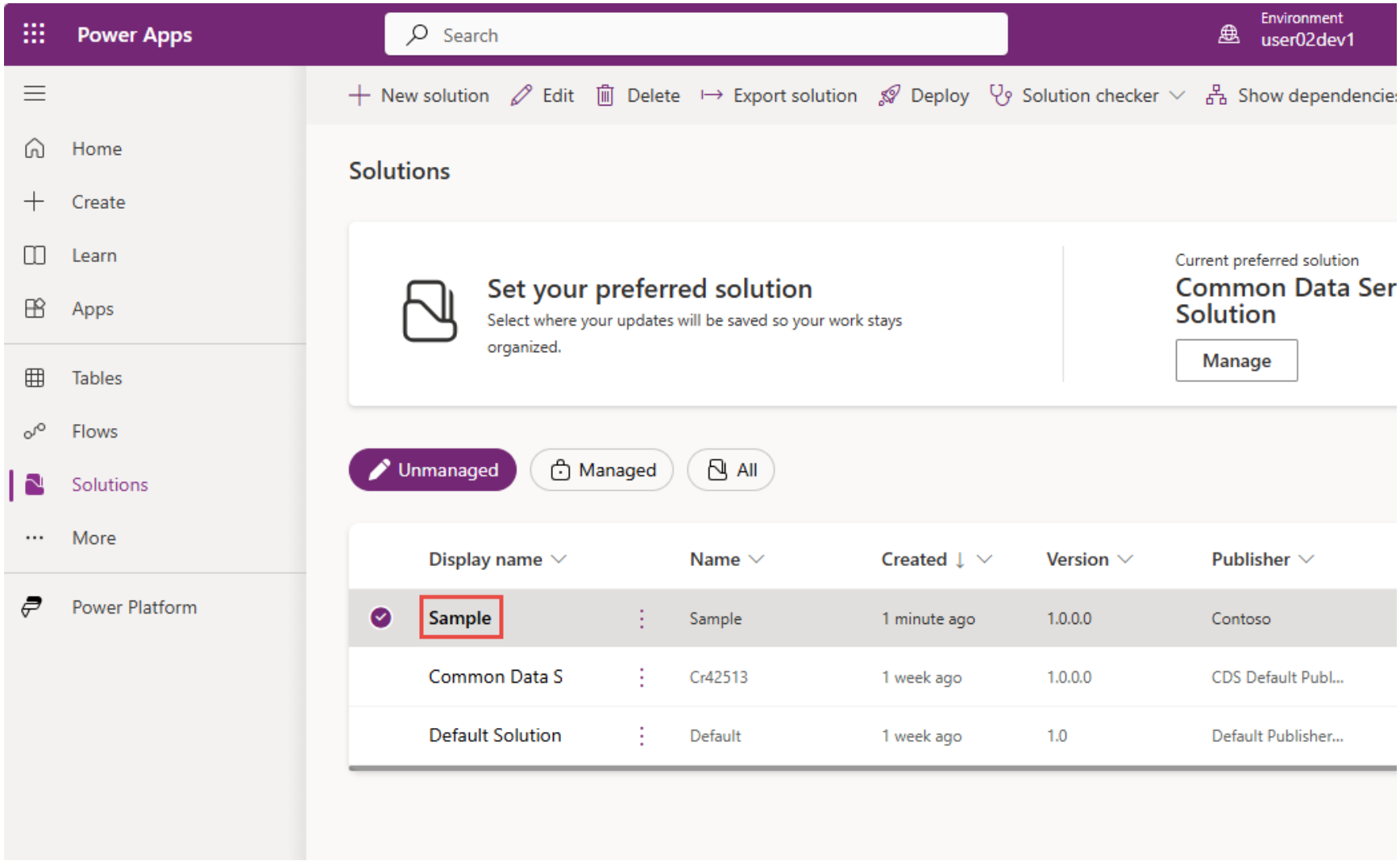
user05_Object

Save the publisher.

Select the publisher record you just created and click the **Create** button to create your solution.

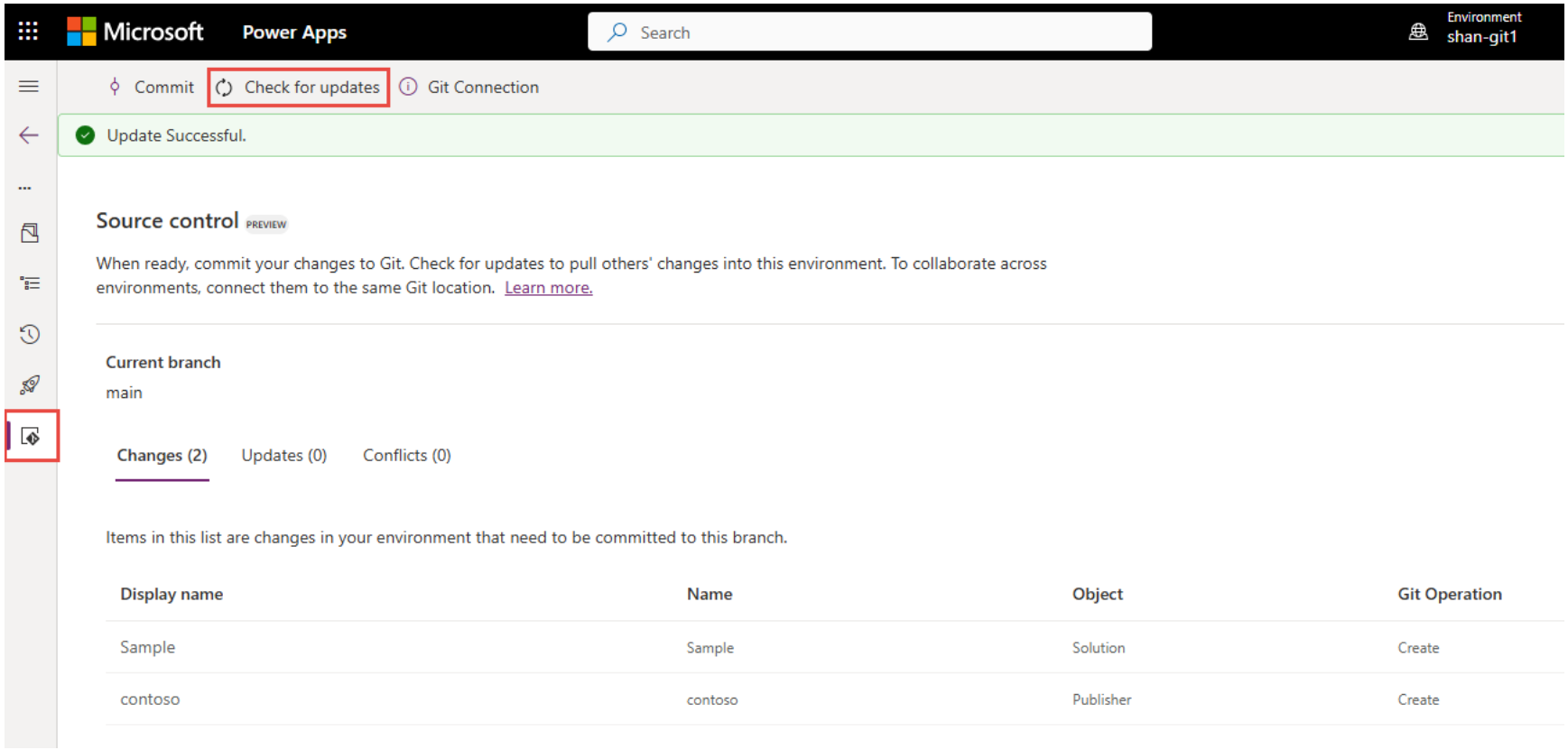


Locate your solution and click to edit it.

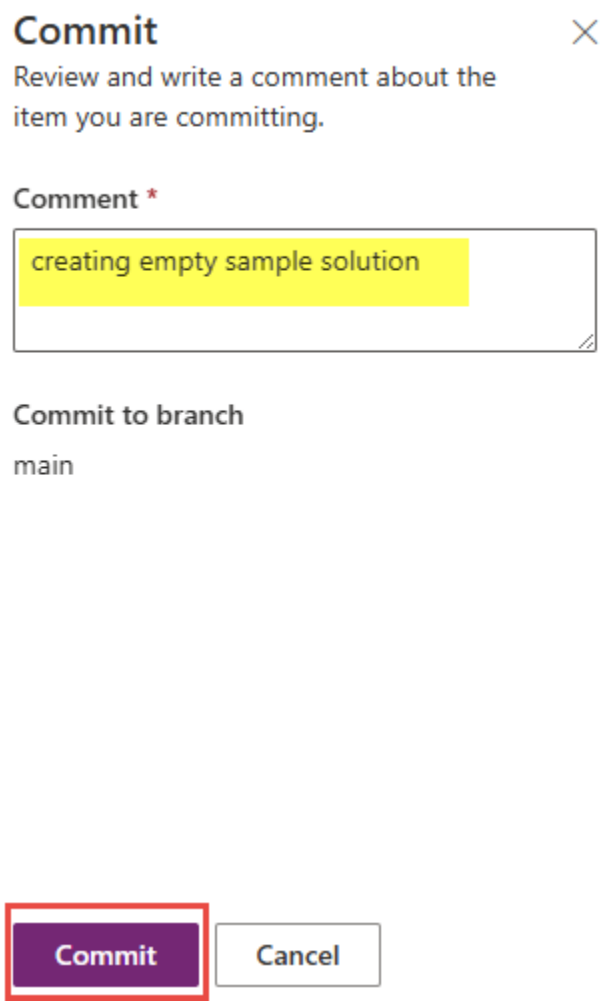


Click on the source control button on the bottom of the left panel, then click the **Check for updates** button in the command bar to refresh the screen. You will see your solution and publisher available to commit to your repository.

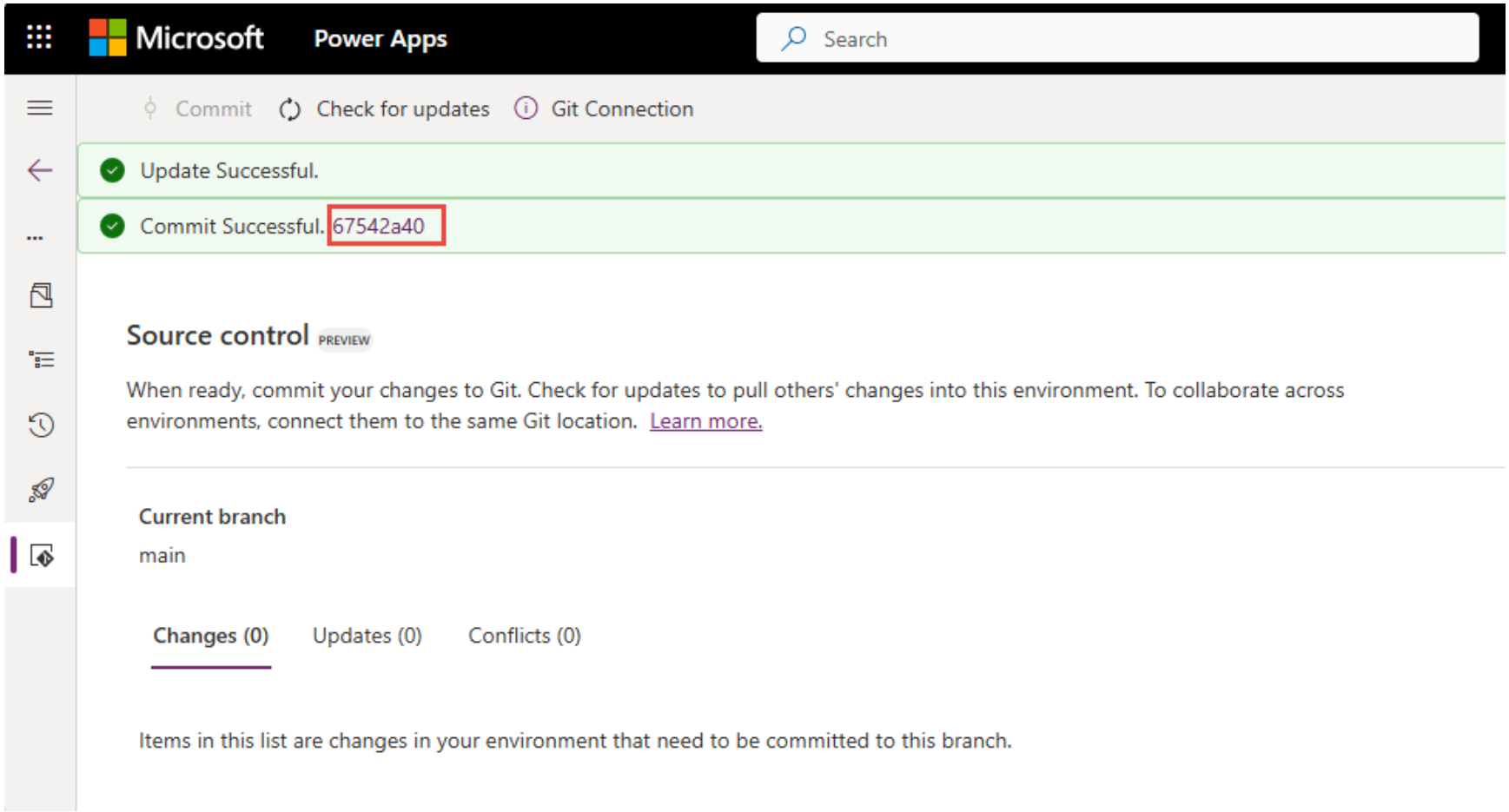
If you don’t see the source control option in the left navigation, make sure you enable the feature using the URL format https://make.preview.powerapps.com?powerappsAlmSourceControl.enableAlmSourceControl=true



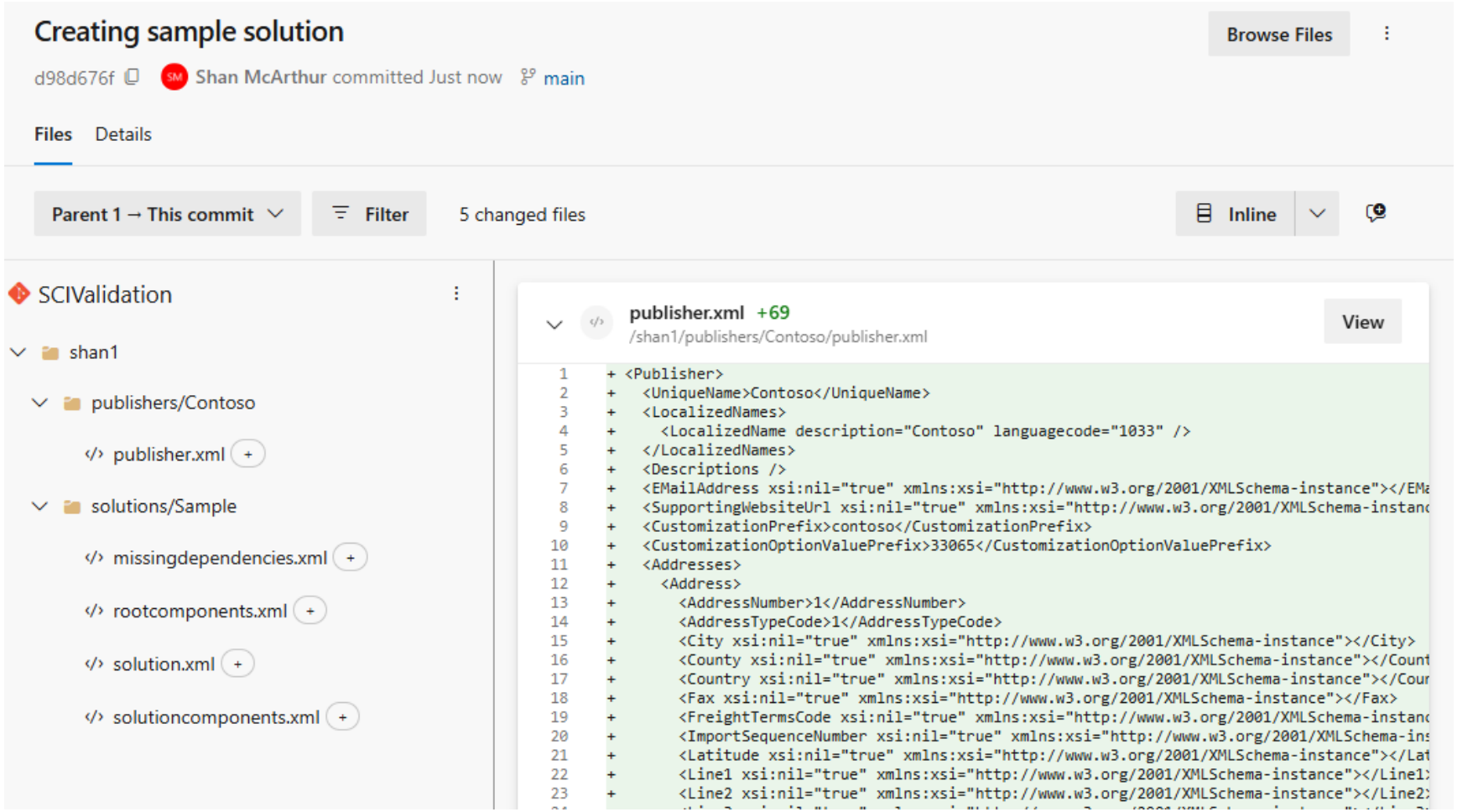
Click the **Commit** button in the command bar and enter a comment and click the **Commit** button to complete the operation.



You will see a new notification on the commit bar when the operation is completed. Click on the commit id to see the details in Azure DevOps.



You can now see the source files that were created in that commit. You should see a new publisher and solution. The solution will be in its own folder and will have a number of different files representing the solution components and dependencies.


Creating sample solution
Browse Files
d98d676f
Shan McArthur committed Just now
main
Files
Details
Parent 1 → This commit
Filter
5 changed files
Inline
SCIValidation
shan1
publishers/Contoso
publisher.xml
solutions/Sample
missingdependencies.xml
rootcomponents.xml
solution.xml
solutioncomponents.xml
publisher.xml +69
/shan1/publishers/Contoso/publisher.xml
View
+ <Publisher>
+ <UniqueName>Contoso</UniqueName>
+ <LocalizedNames>
+ <LocalizedName description="Contoso" languagecode="1033" />
+ </LocalizedNames>
+ <Descriptions />
+ <EMailAddress xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></EMa
+ <SupportingWebsiteUrl xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instanc
+ <CustomizationPrefix>contoso</CustomizationPrefix>
+ <CustomizationOptionValuePrefix>33065</CustomizationOptionValuePrefix>
+ <Addresses>
+ <Address>
+ <AddressNumber>1</AddressNumber>
+ <AddressTypeCode>1</AddressTypeCode>
+ <City xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></City>
+ <County xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></Count
+ <Country xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></Cour
+ <Fax xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></Fax>
+ <FreightTermsCode xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instanc
+ <ImportSequenceNumber xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-ins
+ <Latitude xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></Lat
+ <Line1 xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></Line1
+ <Line2 xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></Line2