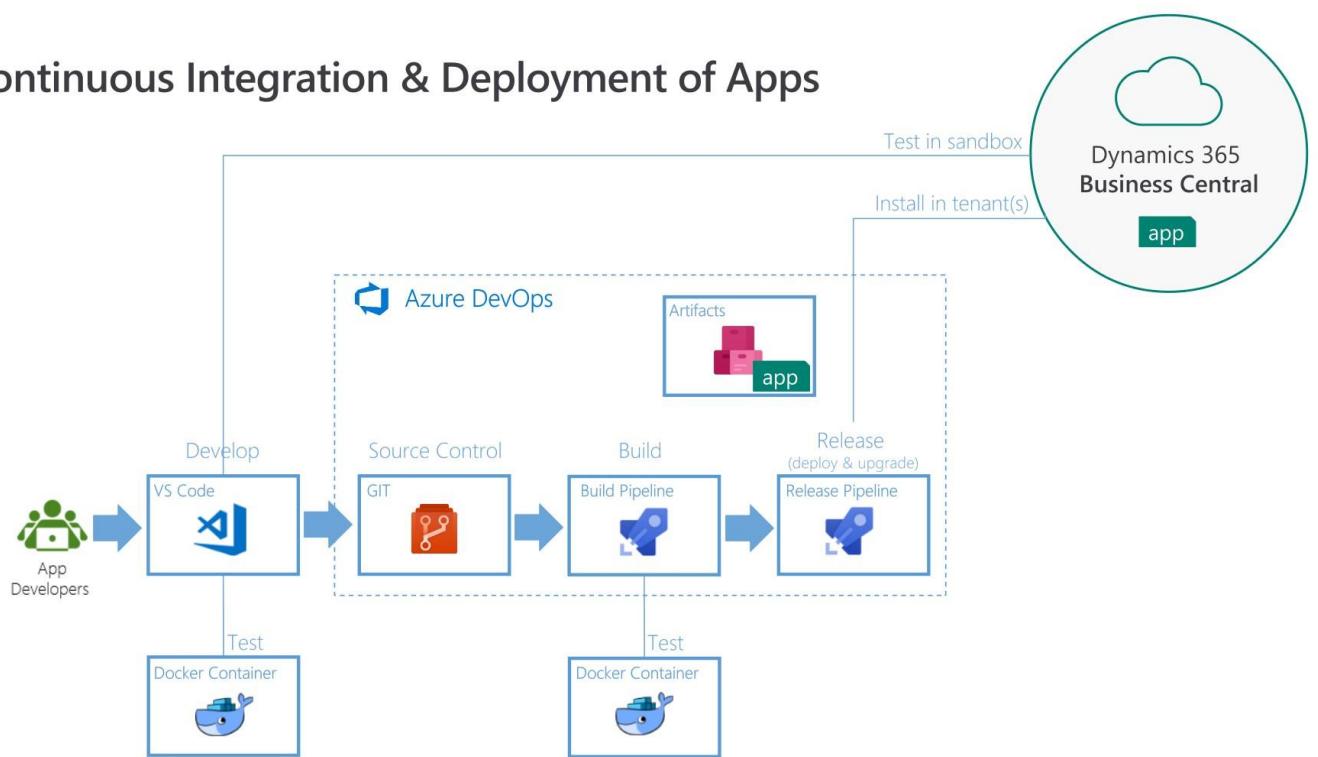


# Azure DevOps – CI/CD Workshop

## Continuous Integration & Deployment of Apps



This workshop will help you setup a project on Azure DevOps, including Continuous Integration, Continuous Deployment and requirements for the developer to submit Pull Requests when checking in.

# Workshop repositories

This workshop uses two repositories. The scripts and pipelines are very similar, but one is a Per Tenant Extension and using the Per Tenant Extension cop + number ranges and the other one is an AppSource app, using the AppSourceCop + appsource number ranges + prefixes and breaking change notifications.

<https://dev.azure.com/businesscentralapps/Old.HelloWorld>

or

<https://github.com/businesscentralapps/Old.HelloWorld>

This is the Per Tenant Extension version of the app.

- Number range defined in app.json is 50100 to 50149.
- PerTenantExtensionCop and UICop are enabled during build (in scripts\settings.json)
- PreviousApps points to .zip file containing previous versions of apps

<https://dev.azure.com/businesscentralapps/Old.HelloWorld.AppSource>

or

<https://github.com/businesscentralapps/Old.HelloWorld.AppSource>

This is the AppSource version of the app.

- Number range defined in app.json is 70074169 to 70074218 (my allocated number range)
- Logo and various URL's in app.json set
- TranslationFile feature enabled
- AppSourceCop and UICop are enabled during build (in scripts\settings.json)
- PreviousApps points to .zip file containing previous versions of apps

The per tenant extension version is the easiest to use for the workshop.

# Workshop environment

To complete this workshop, you need a computer with at least 16Gb of RAM running Windows 10 or Windows Server 2019 with the latest Windows updates applied, the latest Visual Studio Code update and the latest Docker version installed and running Windows Containers.

If your own computer doesn't meet these requirements, you can create a virtual machine on Azure using <http://aka.ms/getbc>. After logging into your Azure subscription, you should **at least** specify valid values for these properties:

- Resource group: <name of resource group>
- Vm Name: <name of VM>
- Accept Eula: Yes
- Remote Desktop Access: \* **(to allow all IP addresses to connect to remote desktop)**
- Admin Password: <my admin password>
- Artifact Url: **bcartifacts/sandbox//us/latest**
- License File Uri: <secure url to a developer/training licensefile>
- Final Setup Script Url: **additional-install.ps1 (or full url to script)**
- Contact E Mail for Let's Encrypt: <my email address>
- Add Traefik: Yes **(if you want to be able to access containers in the VM from the internet)**

Read this blog post <https://freddysblog.com/2017/02/26/create-a-secure-url-to-a-file/> to learn more about how to create a secure url.

**Note** the Final Setup Script Url, which will invoke the script [additional-install.ps1](#) script after the VM is final, which will install chocolatey and use that to install Git, Microsoft Edge, Google Chrome, Firefox and some VS Code extensions.

The remaining properties can be left to their default values. Accept the terms and conditions and press **Purchase**.

Now, the deployment starts. It will take around ~30 minutes until the VM is ready. You can monitor the deployment status on the landing page (<http://<vmname>.<region>.cloudapp.azure.com>). The Landing page should state: Installation Complete before you continue.

# Prerequisites

## Software

If you are using a workshop environment created by <https://aka.ms/getbc>, you will have this software installed already automatically. If you are using your own computer/laptop, you need to check/install this software:

- BcContainerHelper PowerShell module
- Az PowerShell module
- VS Code
- AL Language Extension
- Git
- Microsoft Edge or Google Chrome

## Get an Azure DevOps Account

The Azure DevOps account and organization is where you will create your projects and store your source code. Open <https://azure.microsoft.com/en-us/services/devops/> to create a free account. You will be able to create public or private projects in Azure DevOps.

## Install and configure GIT

Git is the source code management tool used by Visual Studio Code to connect to your Azure DevOps repository.

If GIT isn't already installed, navigate to <https://www.git-scm.com/download/win> and click the download link to download and install Git. Select Visual Studio Code as Git's default editor during installation Wizard and select to commit as-is and checkout as-is when asked.

Configure your username and email in git by starting a **Command Prompt** and type:

```
git config --global user.name "<your username>"  
git config --global user.email "<your email>"  
git config --global credential.helper manager
```

You should use the same email here, as the one used for your Azure DevOps Account.

## Create a Key vault for your secrets

Secrets belong in key vaults and since this workshop will contain license file secrets, passwords and a shared access signature for insider builds, the key vault is needed.

Navigate to <https://portal.azure.com> and login to your subscription. Click Create a resource and select Key vault. Fill out the values as needed and create the Key Vault.

Navigate to the Key Vault resource and navigate to secrets:

Click Generate/Import and fill out the values:

And press create.

For this workshop you need to create a licenserule secret with a secure url to your license file and a password secret.

In PowerShell, accessing these secrets are very simple using the Az PowerShell module.

First, you need to connect your Windows User to your Azure Account, run:

[Connect-AzAccount](#)

login to your Azure Account and now you can use:

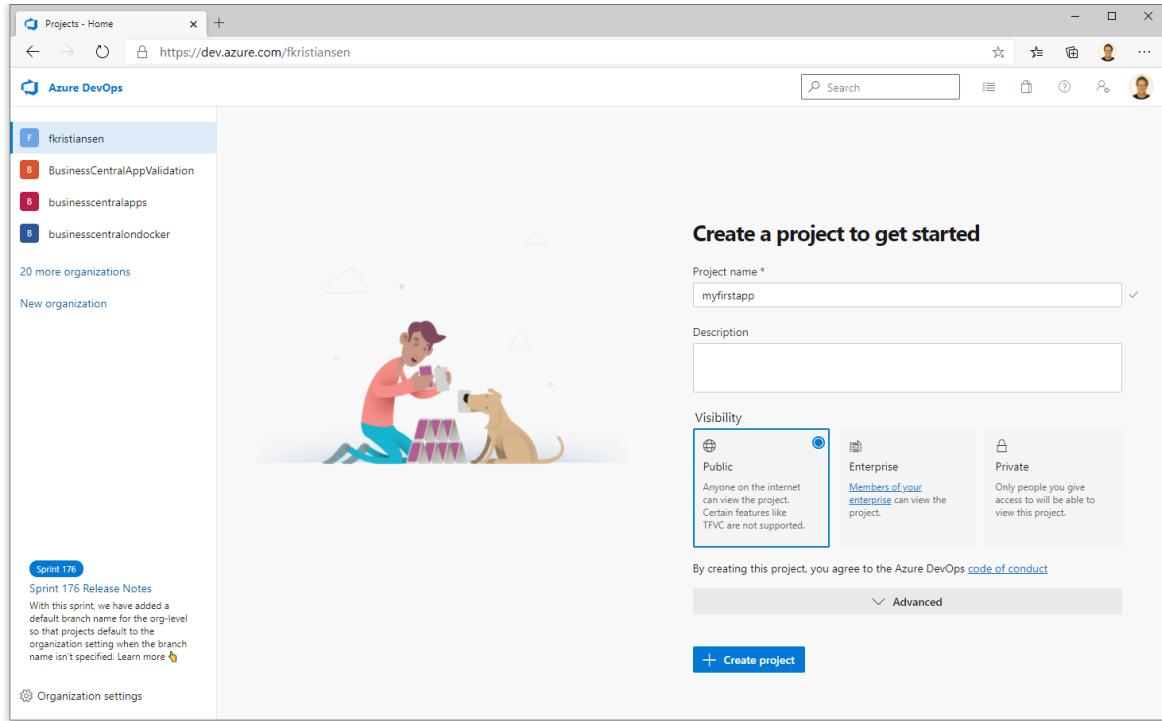
```
$vaultName = "BuildVariables"
$passwordSecret = Get-AzKeyVaultSecret -VaultName $vaultName -Name "Password"
$password = $passwordSecret.SecretValue
```

To read the password as a SecureString. If you want to see the actual password, you need to convert the password to text, which can be done using:

```
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($Password))
```

# Create your organization and your first project

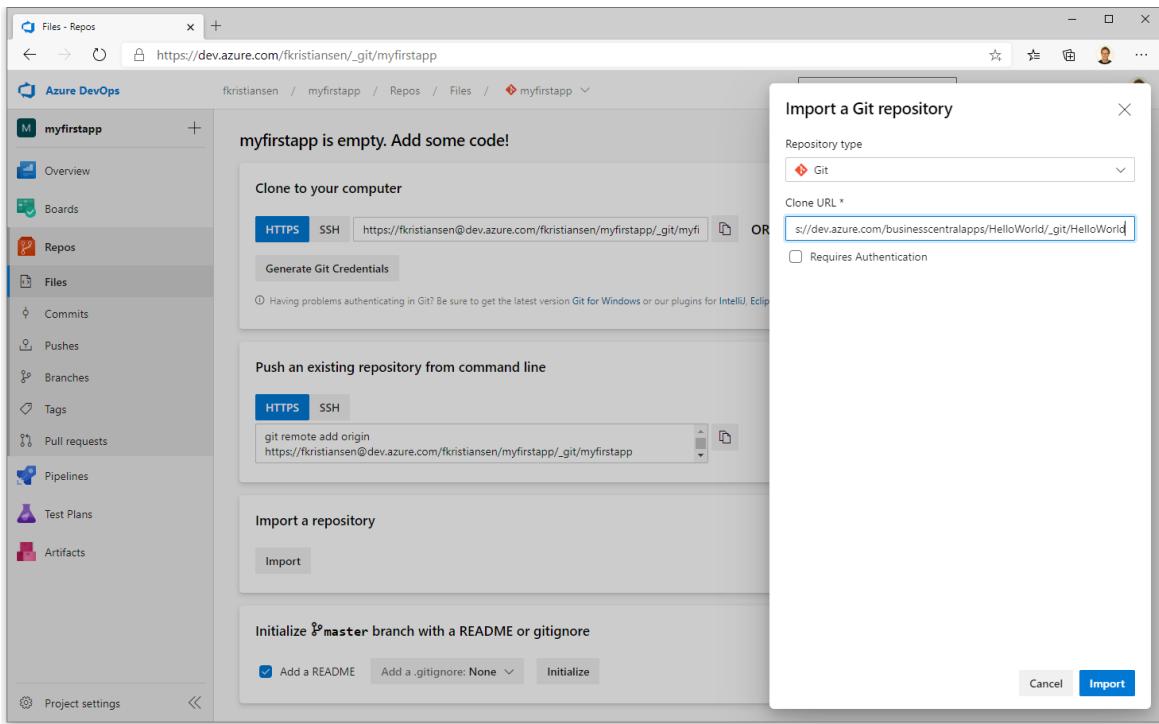
Navigate to <https://devops.azure.com> and login to your DevOps account. Create your organization, which is the location in which you will create your projects. In your organization, create your first project:



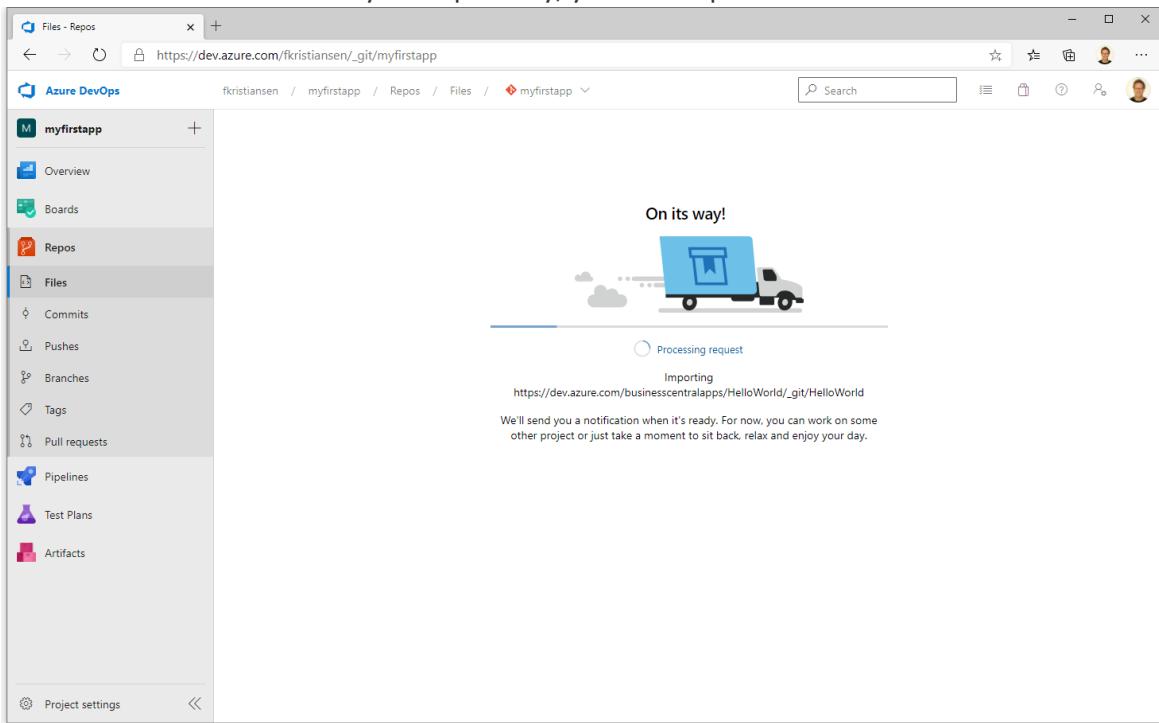
In the project navigate to the **Repos -> Files** area, click **Import** and enter

[https://dev.azure.com/businesscentralapps/Old.HelloWorld/\\_git/Old.HelloWorld](https://dev.azure.com/businesscentralapps/Old.HelloWorld/_git/Old.HelloWorld) (or

Old.HelloWorld.AppSource for the AppSource version) in the Clone URL field. The sample repository is also available on github here: <https://github.com/ BusinessCentralApps/Old.HelloWorld> (add .AppSource for AppSource)



After the truck has delivered your repository, you can inspect the content.



## Inspect the content of the repository

The repository consists of 4 project folders: **base**, **app**, **test** a **scripts** folder.

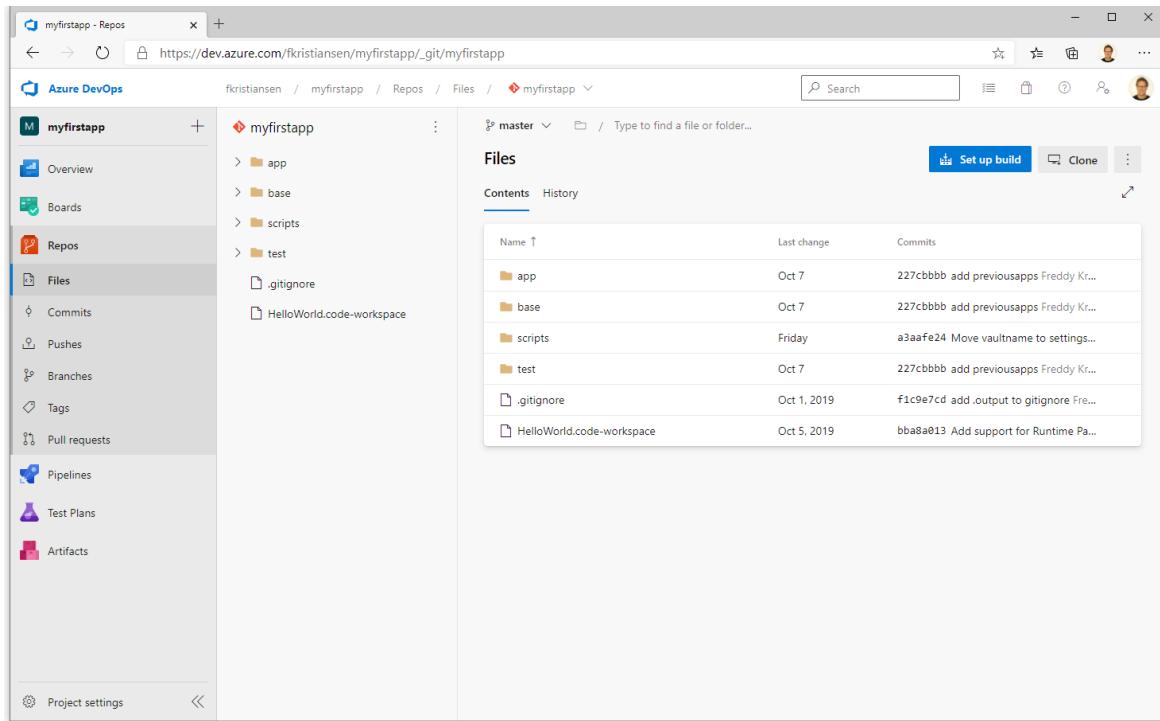
**base** contains a single codeunit with a single function, which returns **App Published: Hello World Base!**

**app** has a dependency to **base** and consists of a **Customer List Page Extension**, which will pop up the Hello World message on the **OnOpenPage Trigger**.

**test** is the test app with a dependency to **app**, containing a **single test**, which opens the **Customer List** and tests that the **Hello World message appears**. **scripts** is a set of scripts/files used for **CI/CD** and setup of dev environments.

The **.gitignore** file is known to everybody who are using GIT as a description of which files GIT should ignore.

The **HelloWorld.code-workspace** is the workspace you want to open with VS Code.



Azure DevOps repository page for myfirstapp. The 'Files' tab is selected, showing the contents of the master branch. The list includes:

Name	Last change	Commits
app	Oct 7	227cbbbb add previousapps Freddy Kr...
base	Oct 7	227cbbbb add previousapps Freddy Kr...
scripts	Friday	a3aafe24 Move vaultname to settings...
test	Oct 7	227cbbbb add previousapps Freddy Kr...
.gitignore	Oct 1, 2019	f1c9e7cd add .output to gitignore Fre...
HelloWorld.code-workspace	Oct 5, 2019	bba8a013 Add support for Runtime Pa...

**Note:** The template will constantly be changed/improved, and the content of the template repository might vary.

If you want to add multiple apps to the project, the idea is to create folders for each app in the root folder.

## Clone the project

In order to work with the project, we need to clone the project to our work machine. You can use the Workshop VM as work machine, or you can use your personal computer/laptop.



Clone repository

Clone Git repository using command line or IDE

Command line

HTTPS SSH

<https://freddykristiansen@dev.azure.com/freddykri...>

Generate Git credentials

IDE

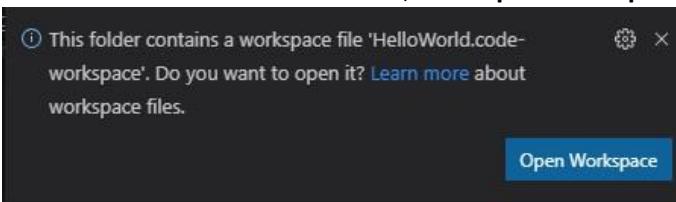
Clone in VS Code

(i) 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](#).

Make sure VS Code is running and click the **Clone** button in the upper right Corner and select **Clone in VS Code**.

Allow the browser to Open VS Code and select a location (f.ex. Documents\AL) for the repository and sign-in to your Azure DevOps account if asked to do so. Say Yes to open the repository.

After opening the repository, VS Code asks whether you want to open the workspace file, click **Open Workspace**.



This folder contains a workspace file 'HelloWorld.code-workspace'. Do you want to open it? [Learn more about workspace files.](#)

Open Workspace

And you should “almost” be ready to start working:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "HELLOWORLD (WORKSPACE)". The "HelloWorld.al" file is selected.
- Code Editor (Top Right):** Displays the content of the "HelloWorld.al" file, which contains AL code for a "CustomerListExt" extension.
- PROBLEMS Panel (Bottom):** Shows several errors related to package caching in the "app.json" and "HelloWorld.al" files.

```
// Welcome to your new AL extension. Freddy Kristiansen, 2 years ago • Initial commit
// Remember that object names and IDs should be unique across all extensions.
// AL snippets start with t*, like tpageext - give them a try and happy coding!
trigger OnOpenPage();
var
    hellobase: Codeunit "Hello Base";
begin
    Message(hellobase.GetText());
end;
}

0 references
pageextension 50101 CustomerListExt extends "Customer List"
{
    trigger OnOpenPage();
    var
        hellobase: Codeunit "Hello Base";
    begin
        Message(hellobase.GetText());
    end;
}
```

PROBLEMS (11) OUTPUT DEBUG CONSOLE TERMINAL

- ① The package cache c:\Users\vmadmin\Documents\AL\myfirstapp\base\.\alpackages could not be found. AL(AL1045) [1, 1]
- ② app.json test (6)
  - ③ The package cache c:\Users\vmadmin\Documents\AL\myfirstapp\test\.\alpackages could not be found. AL(AL1045) [1, 1]
  - ④ The package cache c:\Users\vmadmin\Documents\AL\myfirstapp\test\.\alpackages could not be found. AL(AL1045) [1, 1]
  - ⑤ The package cache c:\Users\vmadmin\Documents\AL\myfirstapp\test\.\alpackages could not be found. AL(AL1045) [1, 1]
  - ⑥ The package cache c:\Users\vmadmin\Documents\AL\myfirstapp\test\.\alpackages could not be found. AL(AL1045) [1, 1]
  - ⑦ The package cache c:\Users\vmadmin\Documents\AL\myfirstapp\test\.\alpackages could not be found. AL(AL1045) [1, 1]
  - ⑧ The package cache c:\Users\vmadmin\Documents\AL\myfirstapp\test\.\alpackages could not be found. AL(AL1045) [1, 1]
- ⑨ HelloWorld.al test (2)
  - ⑩ Page 'Customer List' is missing AL(AL0185) [11, 19]

# Make it your project

The project has been setup with some default **object ids**, **app ids**, **publisher** and **name**. Since this workshop is going to deploy your app to a cloud tenant, you should change these.

You can either fix the solution manually or you can run a small script to fix the solution automatically.

## Fix the solution manually

You can modify app.json in the base project, the app project and the test project manually, setting the app id, app name, app publisher and app versions. Remember that the app project has a dependency on the base project and the test project has a dependency on the app project. Your ids must match. If you are using the AppSource version of HelloWorld, you should also modify the object ids.

## Fix the solution automatically

In VS Code, open the **scripts** folder and open the **MySolution.ps1** file.

File Edit Selection View Go Run Terminal Help MySolution.ps1 - HelloWorld (Workspace) - Visual Studio Code [Administrator]

EXPLORER

> OPEN EDITORS

> HELLOWORLD (WORKSPACE)

> base

> app

> test

> scripts

> .snapshots

! Clmly

> Cleanup.ps1

! CurrentLm

> DevOps-Pipeline.ps1

> Install-BcContainerHelper.ps1

> Local-DevEnv.ps1

> Local-Pipeline.ps1

> MySolution.ps1

! NextMajorLm

! NextMinorLm

> Read-Settings.ps1

! settings.json

> OUTLINE

> TIMELINE

> DOCKER CONTAINERS

> DOCKER IMAGES

> AZURE CONTAINER REGISTRY

> DOCKER HUB

> SUGGESTED DOCKER HUB IMAGES

> Local Sandbox (base)

> MySolution.ps1 <

scripts > MySolution.ps1 < ...

```
1 $path = Join-Path $PSScriptRoot ".."
2
3 $replaceObjectIds = @{
4     "50130" = 80130
5     "50101" = 80101
6     "50100" = 80100
7     "50149" = 80149
8 }
9 $replaceValues = @{
10    "00000000-0000-0000-0000-000000000001" = [Guid]::NewGuid().ToString()
11    "00000000-0000-0000-0000-000000000002" = [Guid]::NewGuid().ToString()
12    "00000000-0000-0000-0000-000000000003" = [Guid]::NewGuid().ToString()
13    "HelloWorld" = "MyApp"
14    "Default Publisher" = "My Name"
15    "Default App Name" = "My App"
16    "Default Base App Name" = "My Base App"
17    "Default Test App Name" = "My Test App"
18    "2.0.0.0" = "1.0.0.0"
19    "https://businesscentralapps.azureedge.net/helloworld/latest/apps.zip" = ""
20 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

3: powershell

Loading personal and system profiles took 2406ms.

PS C:\Users\vmadmin\Documents\VAL\myfirstapp\scripts> .\MySolution.ps1

Modify `$replaceValues` array if needed to indicate the values you want to use on the right side, save the script and run the script. **Note:** Running the script can be tricky, I cannot get F5 to work..., so right-click the PowerShell file and **Open in Integrated Terminal** – or use **Ctrl+Shift+P** and **Open Current File in PowerShell ISE**.

You should see an output indicating which files are modified and if you click the source control symbol, you should see which files have changed and by clicking a file, you can see the changes.

```
app.json (Working Tree) - HelloWorld (Workspace) - Visual Studio Code [Administrator]

File Edi Selection View Go Run Terminal Help app.json (Working Tree) > MySolution.ps1 > app.json (Working Tree) > app.json

Message (Ctrl+Enter to commit on ...)

Changes
app.json app 7
HelloWorldApp 1, M
app.json base 2, M
Codeunit 50130 - Hello Base... M
settings.json scripts 7-
app.json test 8
Codeunit 50133 - HelloWorld T... M

1 "id": "00000000-0000-0000-0000-000000000001",
2 "name": "Default App Name",
3 "publisher": "Default Publisher",
4 "brief": "",
5 "description": "",
6 "version": "2.0.0.0",
7 "privacyStatement": "",
8 "EULA": "",
9 "help": "",
10 "url": "",
11 "logo": "",
12 "dependencies": [
13     {
14         "appId": "00000000-0000-0000-0000-000000000002",
15         "publisher": "Default Publisher",
16         "name": "Default Base App Name",
17         "version": "2.0.0.0"
18     }
19 ],
20 "screenshots": [
21     {
22         "application": "16.0.0.0",
23         "platform": "16.0.0.0",
24         "idRanges": [
25             {
26                 "from": 50100,
27                 "to": 50149
28             }
29         ],
30     }
31 ],
32
33 "application": "16.0.0.0",
34 "platform": "16.0.0.0",
35 "idRanges": [
36     {
37         "from": 80100,
38         "to": 80149
39     }
40 ],
41
42 "id": "f40c4e7c-ae2b-487a-a226-4af7b2c7ea84",
43 "name": "My App",
44 "publisher": "My Name",
45 "brief": "",
46 "description": "",
47 "version": "1.0.0.0",
48 "privacyStatement": "",
49 "EULA": "",
50 "help": "",
51 "url": "",
52 "logo": "",
53 "dependencies": [
54     {
55         "appId": "ff7fiba69-8d08-4745-8113-20",
56         "publisher": "My Name",
57         "name": "My Base App",
58         "version": "1.0.0.0"
59     }
60 ],
61 "screenshots": [
62     {
63         "application": "16.0.0.0",
64         "platform": "16.0.0.0",
65         "idRanges": [
66             {
67                 "from": 80100,
68                 "to": 80149
69             }
70         ],
71     }
72 ],
73
74 "application": "16.0.0.0",
75 "platform": "16.0.0.0",
76 "idRanges": [
77     {
78         "from": 80100,
79         "to": 80149
80     }
81 ],
82
83 "id": "ff7fiba69-8d08-4745-8113-20",
84 "name": "My Base App",
85 "publisher": "My Name",
86 "brief": "",
87 "description": "",
88 "version": "1.0.0.0",
89 "privacyStatement": "",
90 "EULA": "",
91 "help": "",
92 "url": "",
93 "logo": "",
94 "dependencies": [
95     {
96         "appId": "ff7fiba69-8d08-4745-8113-20",
97         "publisher": "My Name",
98         "name": "My Base App",
99         "version": "1.0.0.0"
100     }
101 ],
102 "screenshots": [
103     {
104         "application": "16.0.0.0",
105         "platform": "16.0.0.0",
106         "idRanges": [
107             {
108                 "from": 80100,
109                 "to": 80149
110             }
111         ],
112     }
113 ],
114
115 "application": "16.0.0.0",
116 "platform": "16.0.0.0",
117 "idRanges": [
118     {
119         "from": 80100,
120         "to": 80149
121     }
122 ],
123
124 "id": "ff7fiba69-8d08-4745-8113-20",
125 "name": "My Base App",
126 "publisher": "My Name",
127 "brief": "",
128 "description": "",
129 "version": "1.0.0.0",
130 "privacyStatement": "",
131 "EULA": "",
132 "help": "",
133 "url": "",
134 "logo": "",
135 "dependencies": [
136     {
137         "appId": "ff7fiba69-8d08-4745-8113-20",
138         "publisher": "My Name",
139         "name": "My Base App",
140         "version": "1.0.0.0"
141     }
142 ],
143 "screenshots": [
144     {
145         "application": "16.0.0.0",
146         "platform": "16.0.0.0",
147         "idRanges": [
148             {
149                 "from": 80100,
150                 "to": 80149
151             }
152         ],
153     }
154 ],
155
156 "application": "16.0.0.0",
157 "platform": "16.0.0.0",
158 "idRanges": [
159     {
160         "from": 80100,
161         "to": 80149
162     }
163 ],
164
165 "id": "ff7fiba69-8d08-4745-8113-20",
166 "name": "My Base App",
167 "publisher": "My Name",
168 "brief": "",
169 "description": "",
170 "version": "1.0.0.0",
171 "privacyStatement": "",
172 "EULA": "",
173 "help": "",
174 "url": "",
175 "logo": "",
176 "dependencies": [
177     {
178         "appId": "ff7fiba69-8d08-4745-8113-20",
179         "publisher": "My Name",
180         "name": "My Base App",
181         "version": "1.0.0.0"
182     }
183 ],
184 "screenshots": [
185     {
186         "application": "16.0.0.0",
187         "platform": "16.0.0.0",
188         "idRanges": [
189             {
190                 "from": 80100,
191                 "to": 80149
192             }
193         ],
194     }
195 ],
196
197 "application": "16.0.0.0",
198 "platform": "16.0.0.0",
199 "idRanges": [
200     {
201         "from": 80100,
202         "to": 80149
203     }
204 ],
205
206 "id": "ff7fiba69-8d08-4745-8113-20",
207 "name": "My Base App",
208 "publisher": "My Name",
209 "brief": "",
210 "description": "",
211 "version": "1.0.0.0",
212 "privacyStatement": "",
213 "EULA": "",
214 "help": "",
215 "url": "",
216 "logo": "",
217 "dependencies": [
218     {
219         "appId": "ff7fiba69-8d08-4745-8113-20",
220         "publisher": "My Name",
221         "name": "My Base App",
222         "version": "1.0.0.0"
223     }
224 ],
225 "screenshots": [
226     {
227         "application": "16.0.0.0",
228         "platform": "16.0.0.0",
229         "idRanges": [
230             {
231                 "from": 80100,
232                 "to": 80149
233             }
234         ],
235     }
236 ],
237
238 "application": "16.0.0.0",
239 "platform": "16.0.0.0",
240 "idRanges": [
241     {
242         "from": 80100,
243         "to": 80149
244     }
245 ],
246
247 "id": "ff7fiba69-8d08-4745-8113-20",
248 "name": "My Base App",
249 "publisher": "My Name",
250 "brief": "",
251 "description": "",
252 "version": "1.0.0.0",
253 "privacyStatement": "",
254 "EULA": "",
255 "help": "",
256 "url": "",
257 "logo": "",
258 "dependencies": [
259     {
260         "appId": "ff7fiba69-8d08-4745-8113-20",
261         "publisher": "My Name",
262         "name": "My Base App",
263         "version": "1.0.0.0"
264     }
265 ],
266 "screenshots": [
267     {
268         "application": "16.0.0.0",
269         "platform": "16.0.0.0",
270         "idRanges": [
271             {
272                 "from": 80100,
273                 "to": 80149
274             }
275         ],
276     }
277 ],
278
279 "application": "16.0.0.0",
280 "platform": "16.0.0.0",
281 "idRanges": [
282     {
283         "from": 80100,
284         "to": 80149
285     }
286 ],
287
288 "id": "ff7fiba69-8d08-4745-8113-20",
289 "name": "My Base App",
290 "publisher": "My Name",
291 "brief": "",
292 "description": "",
293 "version": "1.0.0.0",
294 "privacyStatement": "",
295 "EULA": "",
296 "help": "",
297 "url": "",
298 "logo": "",
299 "dependencies": [
300     {
301         "appId": "ff7fiba69-8d08-4745-8113-20",
302         "publisher": "My Name",
303         "name": "My Base App",
304         "version": "1.0.0.0"
305     }
306 ],
307 "screenshots": [
308     {
309         "application": "16.0.0.0",
310         "platform": "16.0.0.0",
311         "idRanges": [
312             {
313                 "from": 80100,
314                 "to": 80149
315             }
316         ],
317     }
318 ],
319
320 "application": "16.0.0.0",
321 "platform": "16.0.0.0",
322 "idRanges": [
323     {
324         "from": 80100,
325         "to": 80149
326     }
327 ],
328
329 "id": "ff7fiba69-8d08-4745-8113-20",
330 "name": "My Base App",
331 "publisher": "My Name",
332 "brief": "",
333 "description": "",
334 "version": "1.0.0.0",
335 "privacyStatement": "",
336 "EULA": "",
337 "help": "",
338 "url": "",
339 "logo": "",
340 "dependencies": [
341     {
342         "appId": "ff7fiba69-8d08-4745-8113-20",
343         "publisher": "My Name",
344         "name": "My Base App",
345         "version": "1.0.0.0"
346     }
347 ],
348 "screenshots": [
349     {
350         "application": "16.0.0.0",
351         "platform": "16.0.0.0",
352         "idRanges": [
353             {
354                 "from": 80100,
355                 "to": 80149
356             }
357         ],
358     }
359 ],
360
361 "application": "16.0.0.0",
362 "platform": "16.0.0.0",
363 "idRanges": [
364     {
365         "from": 80100,
366         "to": 80149
367     }
368 ],
369
370 "id": "ff7fiba69-8d08-4745-8113-20",
371 "name": "My Base App",
372 "publisher": "My Name",
373 "brief": "",
374 "description": "",
375 "version": "1.0.0.0",
376 "privacyStatement": "",
377 "EULA": "",
378 "help": "",
379 "url": "",
380 "logo": "",
381 "dependencies": [
382     {
383         "appId": "ff7fiba69-8d08-4745-8113-20",
384         "publisher": "My Name",
385         "name": "My Base App",
386         "version": "1.0.0.0"
387     }
388 ],
389 "screenshots": [
390     {
391         "application": "16.0.0.0",
392         "platform": "16.0.0.0",
393         "idRanges": [
394             {
395                 "from": 80100,
396                 "to": 80149
397             }
398         ],
399     }
400 ],
401
402 "application": "16.0.0.0",
403 "platform": "16.0.0.0",
404 "idRanges": [
405     {
406         "from": 80100,
407         "to": 80149
408     }
409 ],
410
411 "id": "ff7fiba69-8d08-4745-8113-20",
412 "name": "My Base App",
413 "publisher": "My Name",
414 "brief": "",
415 "description": "",
416 "version": "1.0.0.0",
417 "privacyStatement": "",
418 "EULA": "",
419 "help": "",
420 "url": "",
421 "logo": "",
422 "dependencies": [
423     {
424         "appId": "ff7fiba69-8d08-4745-8113-20",
425         "publisher": "My Name",
426         "name": "My Base App",
427         "version": "1.0.0.0"
428     }
429 ],
430 "screenshots": [
431     {
432         "application": "16.0.0.0",
433         "platform": "16.0.0.0",
434         "idRanges": [
435             {
436                 "from": 80100,
437                 "to": 80149
438             }
439         ],
440     }
441 ],
442
443 "application": "16.0.0.0",
444 "platform": "16.0.0.0",
445 "idRanges": [
446     {
447         "from": 80100,
448         "to": 80149
449     }
450 ],
451
452 "id": "ff7fiba69-8d08-4745-8113-20",
453 "name": "My Base App",
454 "publisher": "My Name",
455 "brief": "",
456 "description": "",
457 "version": "1.0.0.0",
458 "privacyStatement": "",
459 "EULA": "",
460 "help": "",
461 "url": "",
462 "logo": "",
463 "dependencies": [
464     {
465         "appId": "ff7fiba69-8d08-4745-8113-20",
466         "publisher": "My Name",
467         "name": "My Base App",
468         "version": "1.0.0.0"
469     }
470 ],
471 "screenshots": [
472     {
473         "application": "16.0.0.0",
474         "platform": "16.0.0.0",
475         "idRanges": [
476             {
477                 "from": 80100,
478                 "to": 80149
479             }
480         ],
481     }
482 ],
483
484 "application": "16.0.0.0",
485 "platform": "16.0.0.0",
486 "idRanges": [
487     {
488         "from": 80100,
489         "to": 80149
490     }
491 ],
492
493 "id": "ff7fiba69-8d08-4745-8113-20",
494 "name": "My Base App",
495 "publisher": "My Name",
496 "brief": "",
497 "description": "",
498 "version": "1.0.0.0",
499 "privacyStatement": "",
500 "EULA": "",
501 "help": "",
502 "url": "",
503 "logo": "",
504 "dependencies": [
505     {
506         "appId": "ff7fiba69-8d08-4745-8113-20",
507         "publisher": "My Name",
508         "name": "My Base App",
509         "version": "1.0.0.0"
510     }
511 ],
512 "screenshots": [
513     {
514         "application": "16.0.0.0",
515         "platform": "16.0.0.0",
516         "idRanges": [
517             {
518                 "from": 80100,
519                 "to": 80149
520             }
521         ],
522     }
523 ],
524
525 "application": "16.0.0.0",
526 "platform": "16.0.0.0",
527 "idRanges": [
528     {
529         "from": 80100,
530         "to": 80149
531     }
532 ],
533
534 "id": "ff7fiba69-8d08-4745-8113-20",
535 "name": "My Base App",
536 "publisher": "My Name",
537 "brief": "",
538 "description": "",
539 "version": "1.0.0.0",
540 "privacyStatement": "",
541 "EULA": "",
542 "help": "",
543 "url": "",
544 "logo": "",
545 "dependencies": [
546     {
547         "appId": "ff7fiba69-8d08-4745-8113-20",
548         "publisher": "My Name",
549         "name": "My Base App",
550         "version": "1.0.0.0"
551     }
552 ],
553 "screenshots": [
554     {
555         "application": "16.0.0.0",
556         "platform": "16.0.0.0",
557         "idRanges": [
558             {
559                 "from": 80100,
560                 "to": 80149
561             }
562         ],
563     }
564 ],
565
566 "application": "16.0.0.0",
567 "platform": "16.0.0.0",
568 "idRanges": [
569     {
570         "from": 80100,
571         "to": 80149
572     }
573 ],
574
575 "id": "ff7fiba69-8d08-4745-8113-20",
576 "name": "My Base App",
577 "publisher": "My Name",
578 "brief": "",
579 "description": "",
580 "version": "1.0.0.0",
581 "privacyStatement": "",
582 "EULA": "",
583 "help": "",
584 "url": "",
585 "logo": "",
586 "dependencies": [
587     {
588         "appId": "ff7fiba69-8d08-4745-8113-20",
589         "publisher": "My Name",
590         "name": "My Base App",
591         "version": "1.0.0.0"
592     }
593 ],
594 "screenshots": [
595     {
596         "application": "16.0.0.0",
597         "platform": "16.0.0.0",
598         "idRanges": [
599             {
600                 "from": 80100,
601                 "to": 80149
602             }
603         ],
604     }
605 ],
606
607 "application": "16.0.0.0",
608 "platform": "16.0.0.0",
609 "idRanges": [
610     {
611         "from": 80100,
612         "to": 80149
613     }
614 ],
615
616 "id": "ff7fiba69-8d08-4745-8113-20",
617 "name": "My Base App",
618 "publisher": "My Name",
619 "brief": "",
620 "description": "",
621 "version": "1.0.0.0",
622 "privacyStatement": "",
623 "EULA": "",
624 "help": "",
625 "url": "",
626 "logo": "",
627 "dependencies": [
628     {
629         "appId": "ff7fiba69-8d08-4745-8113-20",
630         "publisher": "My Name",
631         "name": "My Base App",
632         "version": "1.0.0.0"
633     }
634 ],
635 "screenshots": [
636     {
637         "application": "16.0.0.0",
638         "platform": "16.0.0.0",
639         "idRanges": [
640             {
641                 "from": 80100,
642                 "to": 80149
643             }
644         ],
645     }
646 ],
647
648 "application": "16.0.0.0",
649 "platform": "16.0.0.0",
650 "idRanges": [
651     {
652         "from": 80100,
653         "to": 80149
654     }
655 ],
656
657 "id": "ff7fiba69-8d08-4745-8113-20",
658 "name": "My Base App",
659 "publisher": "My Name",
660 "brief": "",
661 "description": "",
662 "version": "1.0.0.0",
663 "privacyStatement": "",
664 "EULA": "",
665 "help": "",
666 "url": "",
667 "logo": "",
668 "dependencies": [
669     {
670         "appId": "ff7fiba69-8d08-4745-8113-20",
671         "publisher": "My Name",
672         "name": "My Base App",
673         "version": "1.0.0.0"
674     }
675 ],
676 "screenshots": [
677     {
678         "application": "16.0.0.0",
679         "platform": "16.0.0.0",
680         "idRanges": [
681             {
682                 "from": 80100,
683                 "to": 80149
684             }
685         ],
686     }
687 ],
688
689 "application": "16.0.0.0",
690 "platform": "16.0.0.0",
691 "idRanges": [
692     {
693         "from": 80100,
694         "to": 80149
695     }
696 ],
697
698 "id": "ff7fiba69-8d08-4745-8113-20",
699 "name": "My Base App",
700 "publisher": "My Name",
701 "brief": "",
702 "description": "",
703 "version": "1.0.0.0",
704 "privacyStatement": "",
705 "EULA": "",
706 "help": "",
707 "url": "",
708 "logo": "",
709 "dependencies": [
710     {
711         "appId": "ff7fiba69-8d08-4745-8113-20",
712         "publisher": "My Name",
713         "name": "My Base App",
714         "version": "1.0.0.0"
715     }
716 ],
717 "screenshots": [
718     {
719         "application": "16.0.0.0",
720         "platform": "16.0.0.0",
721         "idRanges": [
722             {
723                 "from": 80100,
724                 "to": 80149
725             }
726         ],
727     }
728 ],
729
730 "application": "16.0.0.0",
731 "platform": "16.0.0.0",
732 "idRanges": [
733     {
734         "from": 80100,
735         "to": 80149
736     }
737 ],
738
739 "id": "ff7fiba69-8d08-4745-8113-20",
740 "name": "My Base App",
741 "publisher": "My Name",
742 "brief": "",
743 "description": "",
744 "version": "1.0.0.0",
745 "privacyStatement": "",
746 "EULA": "",
747 "help": "",
748 "url": "",
749 "logo": "",
750 "dependencies": [
751     {
752         "appId": "ff7fiba69-8d08-4745-8113-20",
753         "publisher": "My Name",
754         "name": "My Base App",
755         "version": "1.0.0.0"
756     }
757 ],
758 "screenshots": [
759     {
760         "application": "16.0.0.0",
761         "platform": "16.0.0.0",
762         "idRanges": [
763             {
764                 "from": 80100,
765                 "to": 80149
766             }
767         ],
768     }
769 ],
770
771 "application": "16.0.0.0",
772 "platform": "16.0.0.0",
773 "idRanges": [
774     {
775         "from": 80100,
776         "to": 80149
777     }
778 ],
779
780 "id": "ff7fiba69-8d08-4745-8113-20",
781 "name": "My Base App",
782 "publisher": "My Name",
783 "brief": "",
784 "description": "",
785 "version": "1.0.0.0",
786 "privacyStatement": "",
787 "EULA": "",
788 "help": "",
789 "url": "",
790 "logo": "",
791 "dependencies": [
792     {
793         "appId": "ff7fiba69-8d08-4745-8113-20",
794         "publisher": "My Name",
795         "name": "My Base App",
796         "version": "1.0.0.0"
797     }
798 ],
799 "screenshots": [
800     {
801         "application": "16.0.0.0",
802         "platform": "16.0.0.0",
803         "idRanges": [
804             {
805                 "from": 80100,
806                 "to": 80149
807             }
808         ],
809     }
810 ],
811
812 "application": "16.0.0.0",
813 "platform": "16.0.0.0",
814 "idRanges": [
815     {
816         "from": 80100,
817         "to": 80149
818     }
819 ],
820
821 "id": "ff7fiba69-8d08-4745-8113-20",
822 "name": "My Base App",
823 "publisher": "My Name",
824 "brief": "",
825 "description": "",
826 "version": "1.0.0.0",
827 "privacyStatement": "",
828 "EULA": "",
829 "help": "",
830 "url": "",
831 "logo": "",
832 "dependencies": [
833     {
834         "appId": "ff7fiba69-8d08-4745-8113-20",
835         "publisher": "My Name",
836         "name": "My Base App",
837         "version": "1.0.0.0"
838     }
839 ],
840 "screenshots": [
841     {
842         "application": "16.0.0.0",
843         "platform": "16.0.0.0",
844         "idRanges": [
845             {
846                 "from": 80100,
847                 "to": 80149
848             }
849         ],
850     }
851 ],
852
853 "application": "16.0.0.0",
854 "platform": "16.0.0.0",
855 "idRanges": [
856     {
857         "from": 80100,
858         "to": 80149
859     }
860 ],
861
862 "id": "ff7fiba69-8d08-4745-8113-20",
863 "name": "My Base App",
864 "publisher": "My Name",
865 "brief": "",
866 "description": "",
867 "version": "1.0.0.0",
868 "privacyStatement": "",
869 "EULA": "",
870 "help": "",
871 "url": "",
872 "logo": "",
873 "dependencies": [
874     {
875         "appId": "ff7fiba69-8d08-4745-8113-20",
876         "publisher": "My Name",
877         "name": "My Base App",
878         "version": "1.0.0.0"
879     }
880 ],
881 "screenshots": [
882     {
883         "application": "16.0.0.0",
884         "platform": "16.0.0.0",
885         "idRanges": [
886             {
887                 "from": 80100,
888                 "to": 80149
889             }
890         ],
891     }
892 ],
893
894 "application": "16.0.0.0",
895 "platform": "16.0.0.0",
896 "idRanges": [
897     {
898         "from": 80100,
899         "to": 80149
900     }
901 ],
902
903 "id": "ff7fiba69-8d08-4745-8113-20",
904 "name": "My Base App",
905 "publisher": "My Name",
906 "brief": "",
907 "description": "",
908 "version": "1.0.0.0",
909 "privacyStatement": "",
910 "EULA": "",
911 "help": "",
912 "url": "",
913 "logo": "",
914 "dependencies": [
915     {
916         "appId": "ff7fiba69-8d08-4745-8113-20",
917         "publisher": "My Name",
918         "name": "My Base App",
919         "version": "1.0.0.0"
920     }
921 ],
922 "screenshots": [
923     {
924         "application": "16.0.0.0",
925         "platform": "16.0.0.0",
926         "idRanges": [
927             {
928                 "from": 80100,
929                 "to": 80149
930             }
931         ],
932     }
933 ],
934
935 "application": "16.0.0.0",
936 "platform": "16.0.0.0",
937 "idRanges": [
938     {
939         "from": 80100,
940         "to": 80149
941     }
942 ],
943
944 "id": "ff7fiba69-8d08-4745-8113-20",
945 "name": "My Base App",
946 "publisher": "My Name",
947 "brief": "",
948 "description": "",
949 "version": "1.0.0.0",
950 "privacyStatement": "",
951 "EULA": "",
952 "help": "",
953 "url": "",
954 "logo": "",
955 "dependencies": [
956     {
957         "appId": "ff7fiba69-8d08-4745-8113-20",
958         "publisher": "My Name",
959         "name": "My Base App",
960         "version": "1.0.0.0"
961     }
962 ],
963 "screenshots": [
964     {
965         "application": "16.0.0.0",
966         "platform": "16.0.0.0",
967         "idRanges": [
968             {
969                 "from": 80100,
970                 "to": 80149
971             }
972         ],
973     }
974 ],
975
976 "application": "16.0.0.0",
977 "platform": "16.0.0.0",
978 "idRanges": [
979     {
980         "from": 80100,
981         "to": 80149
982     }
983 ],
984
985 "id": "ff7fiba69-8d08-4745-8113-20",
986 "name": "My Base App",
987 "publisher": "My Name",
988 "brief": "",
989 "description": "",
990 "version": "1.0.0.0",
991 "privacyStatement": "",
992 "EULA": "",
993 "help": "",
994 "url": "",
995 "logo": "",
996 "dependencies": [
997     {
998         "appId": "ff7fiba69-8d08-4745-8113-20",
999         "publisher": "My Name",
1000         "name": "My Base App",
1001         "version": "1.0.0.0"
1002     }
1003 ],
1004 "screenshots": [
1005     {
1006         "application": "16.0.0.0",
1007         "platform": "16.0.0.0",
1008         "idRanges": [
1009             {
1010                 "from": 80100,
1011                 "to": 80149
1012             }
1013         ],
1014     }
1015 ],
1016
1017 "application": "16.0.0.0",
1018 "platform": "16.0.0.0",
1019 "idRanges": [
1020     {
1021         "from": 80100,
1022         "to": 80149
1023     }
1024 ],
1025
1026 "id": "ff7fiba69-8d08-4745-8113-20",
1027 "name": "My Base App",
1028 "publisher": "My Name",
1029 "brief": "",
1030 "description": "",
1031 "version": "1.0.0.0",
1032 "privacyStatement": "",
1033 "EULA": "",
1034 "help": "",
1035 "url": "",
1036 "logo": "",
1037 "dependencies": [
1038     {
1039         "appId": "ff7fiba69-8d08-4745-8113-20",
1040         "publisher": "My Name",
1041         "name": "My Base App",
1042         "version": "1.0.0.0"
1043     }
1044 ],
1045 "screenshots": [
1046     {
1047         "application": "16.0.0.0",
1048         "platform": "16.0.0.0",
1049         "idRanges": [
1050             {
1051                 "from": 80100,
1052                 "to": 80149
1053             }
1054         ],
1055     }
1056 ],
1057
1058 "application": "16.0.0.0",
1059 "platform": "16.0.0.0",
1060 "idRanges": [
1061     {
1062         "from": 80100,
1063         "to": 80149
1064     }
1065 ],
1066
1067 "id": "ff7fiba69-8d08-4745-8113-20",
1068 "name": "My Base App",
1069 "publisher": "My Name",
1070 "brief": "",
1071 "description": "",
1072 "version": "1.0.0.0",
1073 "privacyStatement": "",
1074 "EULA": "",
1075 "help": "",
1076 "url": "",
1077 "logo": "",
1078 "dependencies": [
1079     {
1080         "appId": "ff7fiba69-8d08-4745-8113-20",
1081         "publisher": "My Name",
1082         "name": "My Base App",
1083         "version": "1.0.0.0"
1084     }
1085 ],
1086 "screenshots": [
1087     {
1088         "application": "16.0.0.0",
1089         "platform": "16.0.0.0",
1090         "idRanges": [
1091             {
1092                 "from": 80100,
1093                 "to": 80149
1094             }
1095         ],
1096     }
1097 ],
1098
1099 "application": "16.0.0.0",
1100 "platform": "16.0.0.0",
1101 "idRanges": [
1102     {
1103         "from": 80100,
1104         "to": 80149
1105     }
1106 ],
1107
1108 "id": "ff7fiba69-8d08-4745-8113-20",
1109 "name": "My Base App",
1110 "publisher": "My Name",
1111 "brief": "",
1112 "description": "",
1113 "version": "1.0.0.0",
1114 "privacyStatement": "",
1115 "EULA": "",
1116 "help": "",
1117 "url": "",
1118 "logo": "",
1119 "dependencies": [
1120     {
1121         "appId": "ff7fiba69-8d08-4745-8113-20",
1122         "publisher": "My Name",
1123         "name": "My Base App",
1124         "version": "1.0.0.0"
1125     }
1126 ],
1127 "screenshots": [
1128     {
1129         "application": "16.0.0.0",
1130         "platform": "16.0.0.0",
1131         "idRanges": [
1132             {
1133                 "from": 80100,
1134                 "to": 80149
1135             }
1136         ],
1137     }
1138 ],
1139
1140 "application": "16.0.0.0",
1141 "platform": "16.0.0.0",
1142 "idRanges": [
1143     {
1144         "from": 80100,
1145         "to": 80149
1146     }
1147 ],
1148
1149 "id": "ff7fiba69-8d08-4745-8113-20",
1150 "name": "My Base App",
1151 "publisher": "My Name",
1152 "brief": "",
1153 "description": "",
1154 "version": "1.0.0.0",
1155 "privacyStatement": "",
1156 "EULA": "",
1157 "help": "",
1158 "url": "",
1159 "logo": "",
1160 "dependencies": [
1161     {
1162         "appId": "ff7fiba69-8d08-4745-8113-20",
1163         "publisher": "My Name",
1164         "name": "My Base App",
1165         "version": "1.0.0.0"
1166     }
1167 ],
1168 "screenshots": [
1169     {
1170         "application": "16.0.0.0",
1171         "platform": "16.0.0.0",
1172         "idRanges": [
1173             {
1174                 "from": 80100,
1175                 "to": 80149
1176             }
1177         ],
1178     }
1179 ],
1180
1181 "application": "16.0.0.0",
1182 "platform": "16.0.0.0",
1183 "idRanges": [
1184     {
1185         "from": 80100,
1186         "to": 80149
1187     }
1188 ],
1189
1190 "id": "ff7fiba69-8d08-4745-8113-20",
1191 "name": "My Base App",
1192 "publisher": "My Name",
1193 "brief": "",
1194 "description": "",
1195 "version": "1.0.0.0",
1196 "privacyStatement": "",
1197 "EULA": "",
1198 "help": "",
1199 "url": "",
1200 "logo": "",
1201 "dependencies": [
1202     {
1203         "appId": "ff7fiba69-8d08-4745-8113-20",

```

## Build the solution manually

To get into a situation where RAD (Rapid Application Development works), you need to create a development environment and build+publish your projects. Starting with the base, you would need to download symbols, compile and publish all apps after creating a container with the right version.

## Build the solution automatically

In VS Code, open the **scripts** folder and open the **Local-DevEnv.ps1** file.

Open the PowerShell terminal and run **Connect-AzAccount** to allow PowerShell access to your Azure Account and **Select-AzSubscription** to get access to the Subscription where your KeyVault was created.

```

File Edit Selection View Go Run Terminal Help Local-DevEnv.ps1 - HelloWorld [Workspace] - Visual Studio Code [Administrator]
EXPLORER Local-DevEnv.ps1 Local-DevEnv.ps1 ...
scripts > Local-DevEnv.ps1 ...
scripts > Local-DevEnv.ps1 ...
You, 2 days ago | You, 5 days ago | add local pipeline and local devenv
1 Param([Parameter(Mandatory=$false)]
2 [string]$version = "v1"
3 )
4
5 $baseFolder = (Get-Item (Join-Path $PSScriptRoot "")).FullName
6 . (Join-Path $PSScriptRoot "Read-Settings.ps1") -version $version
7
8 $bcContainerHelperVersion = "latest"
9 if ($settings.PSObject.Properties.Name -eq 'bcContainerHelperVersion' -and $settings.bcContainerHelperVersion) {
10     $bcContainerHelperVersion = $settings.bcContainerHelperVersion
11 }
12 Write-Host "Use bcContainerHelper Version: $bcContainerHelperVersion"
13 . (Join-Path $PSScriptRoot "Install-BcContainerHelper.ps1") -bcContainerHelperVersion $bcContainerHelperVersion
14
15 if ($genericImageName) {
16     $bcContainerHelperConfig.genericImageName = $genericImageName
17 }
18
19
20 if ("$vaultNameForLocal" -eq "") -or !(Get-AzKeyVault -VaultName $vaultNameForLocal)) {
21     throw "You need to setup a Key Vault for use with local pipelines"
22 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
Loading personal and system profiles took 2487ms.  
PS C:\Users\vmadmin\Documents\VAL\myfirstapp\scripts> connect-azaccount  
PS C:\Users\vmadmin\Documents\VAL\myfirstapp\scripts>

Account SubscriptionName TenantId Environment  
freddyk@microsoft.com Workshops and Demos 72f988bf-86f1-41af-91ab-2d7cd011db47 AzureCloud

PS C:\Users\vmadmin\Documents\VAL\myfirstapp\scripts>

You, 5 days ago | You, 5 days ago | In 1, Col 1 | Spaces: 4 | UTF-8 with BOM | CRLF | PowerShell | 5.1 | 🔍

Now run .\Local-DevEnv.ps1

This script will create a devopment container, compile and publish all apps using the Dev Endpoint and leave the container running, allowing you to modify and publish individual apps afterwards.

**Note:** that the local-devenv script assumes that you have a key vault called **BuildVariables** with at least a **licensefile** and a **password** secret.

You should see the script running, creating the container, compiling and publishing apps and modifying the launch.json file to prepare your VS Code for Rapid Application Development.

Creating container

```

dotnet ContainerManager Create-Container -n Local-DevEnv -t Docker -p Local-DevEnv.ps1
dotnet ContainerManager Start-Container -n Local-DevEnv

```

Compiling apps

```

dotnet ContainerManager Publish-Container -n Local-DevEnv -p Local-DevEnv.ps1

```

```

File Edit Selection View Go Run Terminal Help Local-DevEnv.ps1 - HelloWorld (Workspace) - Visual Studio Code [Administrator]
EXPLORER OPEN EDITORS Local-DevEnv.ps1
scripts > Local-DevEnv.ps1 ...
16 if ($genericImageName) {
17     $bcContainerHelperConfig.genericImageName = $genericImageName
18 }
19
20 if ("$vaultNameForLocal" -eq "") -or !(Get-AzKeyVault -VaultName $vaultNameForLocal)) {
21     throw "You need to setup a Key Vault for use with local pipelines"
22 }
23 Get-AzKeyVaultSecret -VaultName $vaultNameForLocal | ForEach-Object {
24     Write-Host "Get Secret ${_.Name} Secret"
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell
Using Symbols Folder: C:\Users\vmadmin\Documents\AL\myfirstapp\test\alPackages
Disabling SSL Verification
Re-enabling SSL Verification
Compiling...
.alc.exe /project:"c:\sources\test" /packagecachePath:"c:\sources\test\alPackages" /out:"c:\sources\test\My Name_My Test App_1.0.0.0.app" /assemblyprobingsPaths:"C:\Program Files (x86)\Microsoft Dynamics NAV170\RoleTailored Client","C:\Program Files\Microsoft Dynamics NAV\170\Service","C:\Program Files (x86)\Open XML SDK\2.5\lib","C:\Windows\Microsoft.NET\Assembly","C:\Test Assemblies\Mock Assemblies"
Microsoft (R) AL Compiler version 6.0.5.10826
Copyright (C) Microsoft Corporation. All rights reserved

Compilation started for project 'My Test App' containing '1' files at '13:09:47.39'.

Compilation ended at '13:09:57.467'.

C:\Users\vmadmin\Documents\AL\myfirstapp\test\My Name_My Test App_1.0.0.0.app successfully created in 12 seconds
Disabling SSL Verification
Publishing My Name_My Test App_1.0.0.0.app to https://172.17.281.196:7049/BC/dev/apps?schemaUpdateMode=synchronize&tenant=default
Re-enabling SSL Verification
App successfully published
Modifying C:\Users\vmadmin\Documents\AL\myfirstapp\test\vscode\launch.json

Compiling apps, test apps and importing test toolkit took 97 seconds
AL Pipeline finished in 230 seconds
PS C:\Users\vmadmin\Documents\AL\myfirstapp\scripts>

```

You, 2 days ago Ln 22, Col 2 Spaces: 4 UTF-8 with BOM CRLF PowerShell 5.1

Press Ctrl+Shift+P and execute Developer: Reload Window to get rid of cached compiler errors.

```

File Edit Selection View Go Run Terminal Help HelloWorld.al - HelloWorld (Workspace) - Visual Studio Code [Administrator]
EXPLORER OPEN EDITORS HelloWorld.al
app > HelloWorld.al
1 // Welcome to your new AL extension.
2 // Remember that object names and IDs should be unique across all extensions.
3 // AL snippets start with ***, like spageext - give them a try and happy coding!
4
5 pageextension 80101 CustomerListExt extends "Customer List"
6 [
7     trigger OnOpenPage();
8     var
9         hellobase: Codeunit "Hello Base";
10    begin
11        Message(hellobase.GetText());
12    end;
13 ]
14
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
2: PowerShell Integrated Console v2020.6.0 <=====

PS C:\Users\vmadmin\Documents\AL\myfirstapp\base>

```

You, a few seconds ago Ln 12, Col 9 Spaces: 4 UTF-8 CRLF AL

Now you can navigate to HelloWorld.al and press F5.

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows the project structure under "HELLOWORLD (WORKSPACE)".
- Editor View:** Displays the code for `HelloWorld.al`. The code is as follows:

```
You, a few seconds ago | 1 author (You)
1 // Welcome to your new AL extension.
2 // Remember that object names and IDs should be unique across all extensions.
3 // AL snippets start with t*, like tpageext - give them a try and happy coding!
4
5 0 references
6 pageextension 80101 CustomerListExt extends "Customer List"
7
8 {
9     trigger OnOpenPage();
10    var
11        hellobase: Codeunit "Hello Base";
12        begin
13            Message(hellobase.GetText());
14        end;
15 }
```

- Terminal View:** Shows deployment logs from 2020-10-25 at 14:17:34.64 to 14:17:37.63.
- Status Bar:** Shows the current branch as "master".

And then it should automatically launch the Web Client.

The screenshot shows the Dynamics 365 Business Central interface with the following details:

- Header:** Shows the URL <https://fkworkshop.northeurope.cloudapp.azure.com/myapp-ci/?tenant=default&sk=3195da19-7c6c-44bd-bd58-bb9d2dd37e9e&page=22&co...>
- Table View:** Displays a list of customers.
- Modal Dialog:** A central modal window displays the message "App Published: Hello World Base!" with an "OK" button.
- Right Panel:** Shows the "Sell-to Customer Sales History" section with various sales-related metrics.

## Check-in your changes

Typically, we do not check-in **launch.json**, in VS Code, you can add these to **.gitignore** in order to avoid tracking them if you like, in this workshop we will just check in everything.

Click the Source Control icon and press + on the changes line to stage all your changes.

SOURCE CONTROL    lauch.json (Index)    ...

```

initial change
  ✓ Staged Changes (12)
    app.json app M
    HelloWorld.app M
    launch.json app + vscode M
    app.json base M
    Codeunit 50130 - Hello Base.al... M
    launch.json base + vscode M
    Local-DevEnv.ps1 scripts M
    MySolution.ps1 scripts M
    settings.json scripts M
    app.json test M
    Codeunit 50133 - HelloWorld T... M
    launch.json test + vscode M
  ✓ Changes (0)

c:\> Users\vmadmin\Documents\AL>myfirstapp>app>.vscode>launch.json Launch Targets > {} Local Sandbox
1 {
  "version": "0.2.0",
  "configurations": [
    {
      "type": "al",
      "request": "launch",
      "name": "Local Sandbox",
      "server": "http://helloworld-current",
      "serverInstance": "BC",
      "port": 7049,
      "tenant": "default",
      "authentication": "UserPassword",
      "startupObjectId": 22,
      "startupObjectType": "Page",
      "breakOnError": true
    }
  ]
}

1 {
  "version": "0.2.0",
  "configurations": [
    {
      "type": "al",
      "request": "launch",
      "name": "Local Sandbox",
      "server": "https://fkworkshop.northeurope.cloudapp.azure.com",
      "serverInstance": "myapp-cidev",
      "port": 443,
      "tenant": "default",
      "authentication": "UserPassword",
      "startupObjectId": 22,
      "startupObjectType": "Page",
      "breakOnError": true
    }
  ]
}

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
[2020-10-25 15:56:40.21] Publishing AL application using launch configuration 'Local Sandbox'.
[2020-10-25 15:56:40.67] Targeting server 'https://fkworkshop.northeurope.cloudapp.azure.com', server instance 'myapp-cidev' and tenant 'default'.
[2020-10-25 15:56:40.69] Using user name and password authentication. User name used is: 'admin'.
[2020-10-25 15:56:40.71] Sending request to https://fkworkshop.northeurope.cloudapp.azure.com/myapp-cidev/dev/metadata?tenant=default
[2020-10-25 15:56:41.33] Publishing package to tenant 'default'
[2020-10-25 15:56:41.33] Targeting server 'https://fkworkshop.northeurope.cloudapp.azure.com', server instance 'myapp-cidev' and tenant 'default'.
[2020-10-25 15:56:41.33] Using user name and password authentication. User name used is: 'admin'.
[2020-10-25 15:56:41.34] Sending request to https://fkworkshop.northeurope.cloudapp.azure.com/myapp-cidev/dev/apps?tenant=default&schemaUpdateMode=synchronize&dependencyPublishingOption=default
[2020-10-25 15:56:43.50] Success: The package 'My Name_My App_1.0.0.dep.app' has been published to the server.

>
```

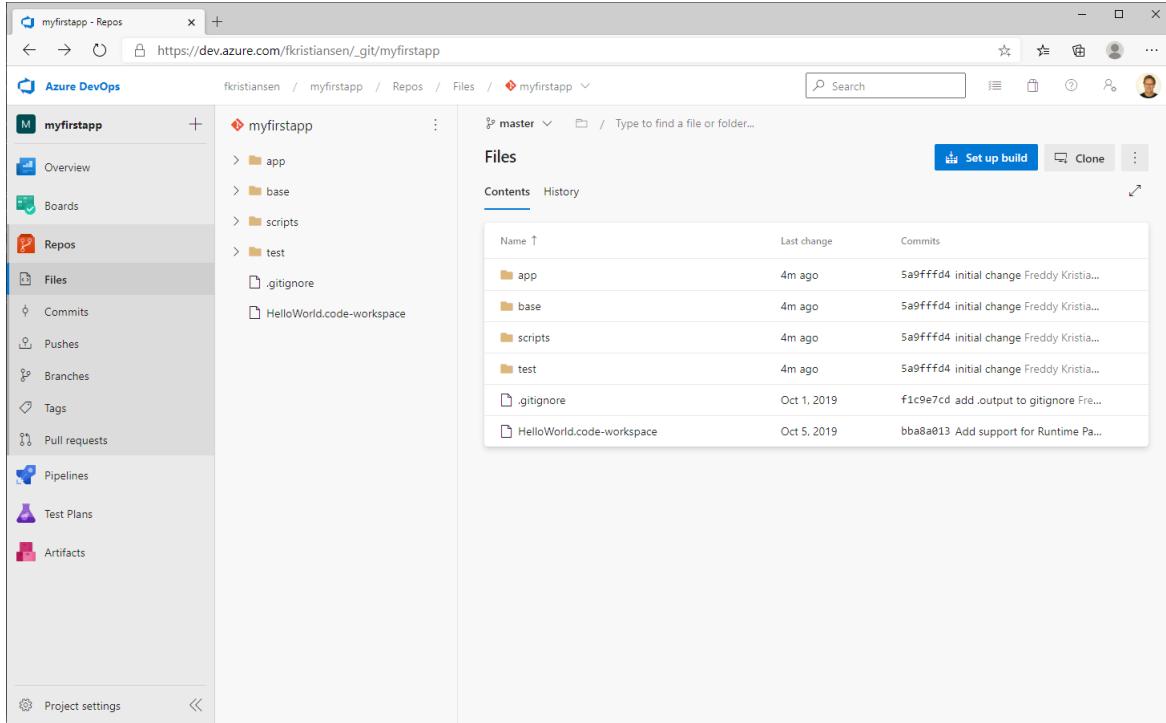
Ln 8, Col 1   Spaces: 4   UTF-8   JSON with Comments

Staged changes are changes you want to commit. Enter a commit message and hit Commit (or use **Ctrl+Enter**). In the bottom left corner, identify the **Synchronize Changes** symbol and hit that. You might be asked to login to your devops account in order to push your changes.



# Create a Build Pipeline

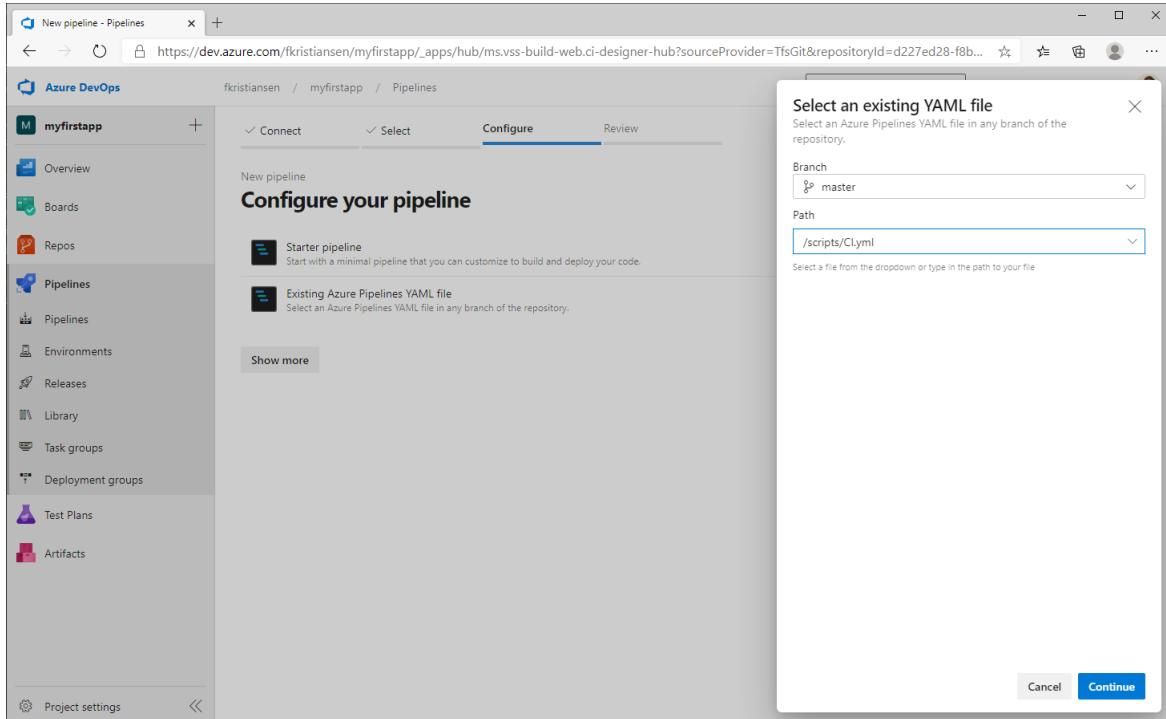
In Azure DevOps, under your Project, **Repos** -> **Files**, you will see a button called **Set up build**. Click it.



The screenshot shows the 'Files' section of the Azure DevOps interface for the 'myfirstapp' repository. The left sidebar includes options like Overview, Boards, Repos (selected), Files, Commits, Pushes, Branches, Tags, Full requests, Pipelines, Test Plans, and Artifacts. The 'Project settings' link is at the bottom. The main area displays a list of files and folders: app, base, scripts, test, .gitignore, and HelloWorld.code-workspace. A 'Set up build' button is located in the top right corner of the file list area.

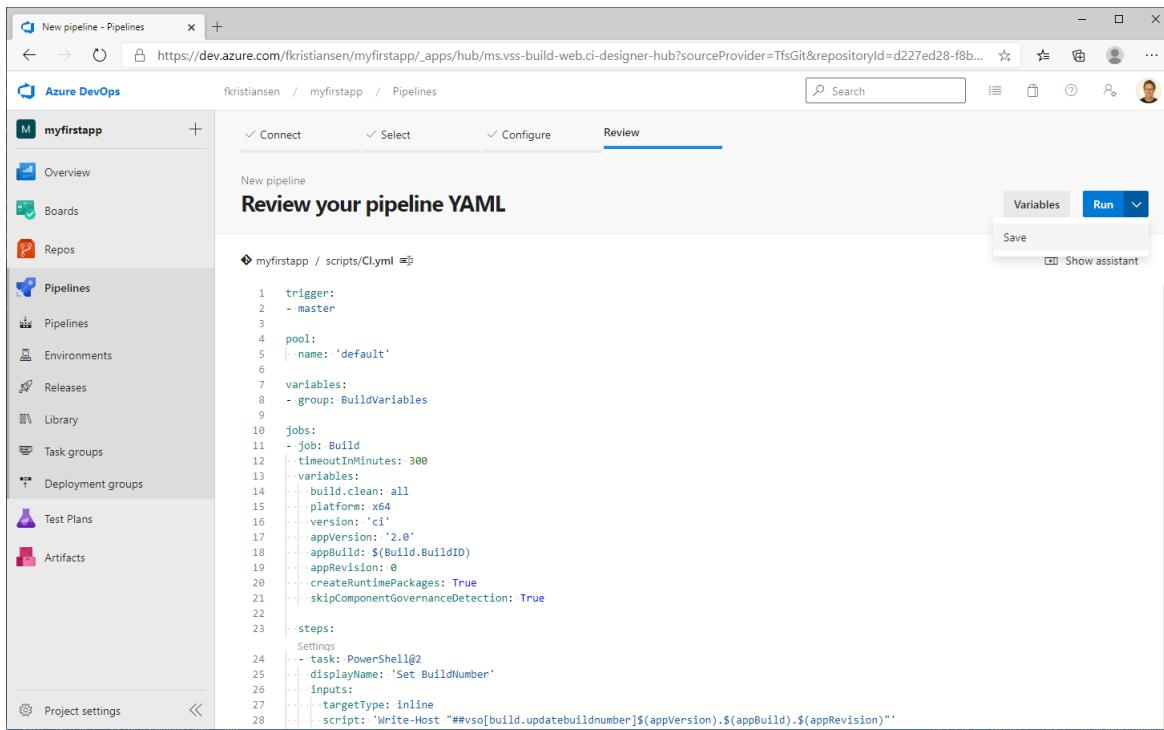
In the **Configure your pipeline**, select **Existing Azure Pipelines YAML file**.

Select **./.azureDevOps/Cl.yml** in the path for the YAML file and press **Continue**.



The screenshot shows the 'Configure your pipeline' step of the pipeline creation wizard. The left sidebar lists Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The main area has tabs for Connect, Select, Configure (selected), and Review. Under 'Configure your pipeline', there are two options: 'Starter pipeline' and 'Existing Azure Pipelines YAML file'. The 'Existing Azure Pipelines YAML file' option is selected, and a modal window titled 'Select an existing YAML file' is displayed. The modal shows a dropdown for 'Branch' set to 'master' and a text input for 'Path' containing '/scripts/Cl.yml'. At the bottom of the modal are 'Cancel' and 'Continue' buttons.

In the **Review your Pipeline YAML**, click the arrow next to the Run button and Save the pipeline.



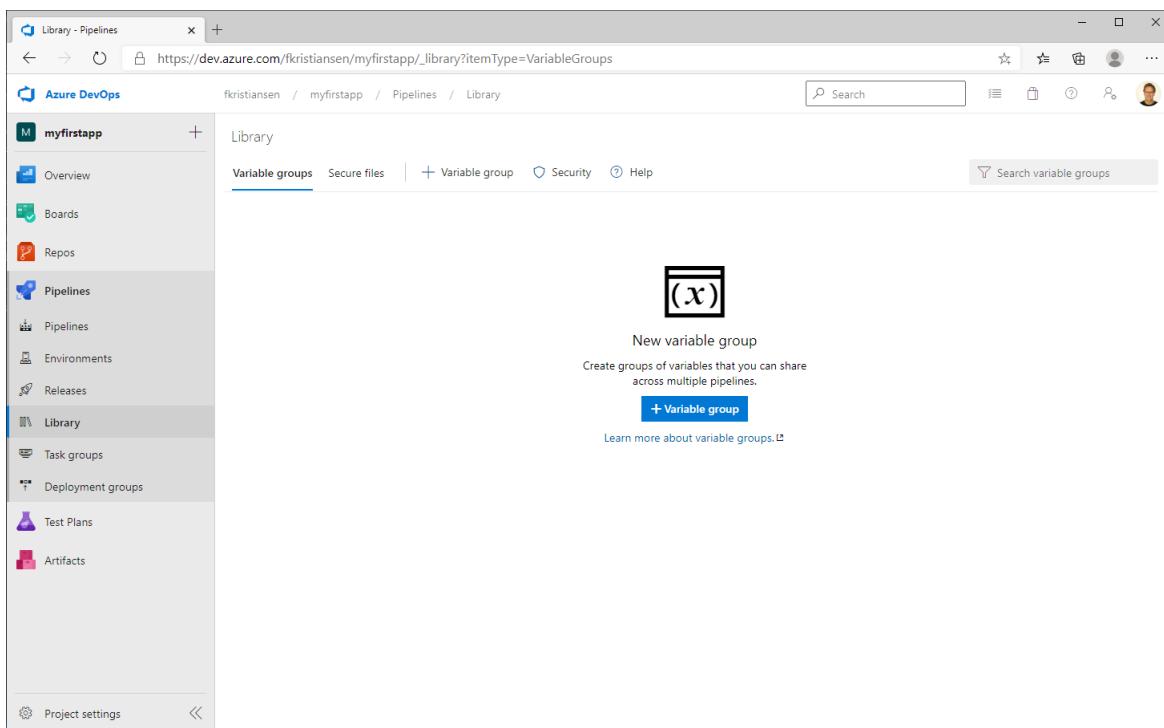
```
trigger:
- master

pool:
  name: 'default'

variables:
- group: BuildVariables

jobs:
- job: Build
  timeoutInMinutes: 300
  variables:
    build.clean: all
    platform: x64
    version: 'ci'
    appVersion: '2.0'
    appBuild: ${Build.BuildID}
    appRevision: 0
    createRuntimePackages: True
    skipComponentGovernanceDetection: True
  steps:
    - task: PowerShell@2
      displayName: 'Set BuildNumber'
      inputs:
        targetType: inline
        script: 'Write-Host "#vso[build.updatebuildnumber]$(appVersion).$(appBuild).$(appRevision)"'
```

The pipeline needs access to your Key Vault secrets. Navigate to Library under Pipelines:



New variable group

Create groups of variables that you can share across multiple pipelines.

+ Variable group

Learn more about variable groups.

Create a variable group called BuildVariables. Link Secrets from an Azure Key Vault as variables and authorize the pipeline to access your subscription and your Key Vault:

The screenshot shows the Azure DevOps interface for creating a variable group. The left sidebar is for project management, and the main area is for 'Library > BuildVariables'. The 'Properties' section includes a 'Variable group name' field set to 'BuildVariables', a 'Description' field, and two toggle buttons: 'Allow access to all pipelines' and 'Link secrets from an Azure key vault as variables'. Below this is a 'Manage' section for 'Azure subscription' and 'Key vault name'. A dropdown menu for 'Key vault name' has 'BuildVariables' selected. A note indicates that secrets will be managed by Azure Pipelines.

After that, add the variables needed by the pipeline:

The screenshot shows the 'Choose secrets' dialog box. It lists secrets to be included in the variable group, with checkboxes for selection. The checked secrets are 'Licensefile' and 'Password'. Other secrets listed include 'InsiderSasToken', 'KeyVaultCertificateFile', 'KeyVaultCertificatePassword', 'KeyVaultClientId', 'privateSasToken', and 'StorageConnectionString'. The dialog has 'Ok' and 'Cancel' buttons at the bottom.

Click Ok and Save!

Go back to pipelines, click your pipeline and select **Run Pipeline**.

The pipeline will ask for permissions to access a resource (your Key Vault) before running. This can either be in the pipeline window, like here:

A screenshot of the Azure DevOps Pipelines summary page. The URL is [https://dev.azure.com/fkristiansen/myfirstapp/\\_build/results?buildId=2&view=results](https://dev.azure.com/fkristiansen/myfirstapp/_build/results?buildId=2&view=results). The pipeline name is 'myfirstapp'. The summary shows it was manually run by 'Freddy Kristiansen' at 'Just now'. It lists 'myfirstapp' as the repository and version, with the master branch checked out. There are 0 work items and 0 artifacts related. A 'View 150 changes' link is available. A warning message states: 'This pipeline needs permission to access a resource before this run can continue'. The 'Jobs' section shows one job named 'Build' in a 'Waiting' state.

or inside the actual build window, like here:

A screenshot of the Azure DevOps Pipeline logs page for run #20201025.2. The URL is [https://dev.azure.com/fkristiansen/myfirstapp/\\_build/results?buildId=2&view=logs&j=6884a131-87da-5381-61f3-d7acc3b91d76](https://dev.azure.com/fkristiansen/myfirstapp/_build/results?buildId=2&view=logs&j=6884a131-87da-5381-61f3-d7acc3b91d76). The pipeline name is 'myfirstapp'. The 'Build' job is listed under 'Jobs' with the status 'Pending'. A warning message states: 'This pipeline needs permission to access a resource before this run can continue'. A 'View' button is present.

**Click view and grant permissions.**

Follow the progress of the pipeline by clicking the pipeline:

```

83  Test application folders
84  - D:\a\1\s\test
85
86
87  / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \
88  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
89  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
90  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
91  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
92  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
93  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
94
95 Pulling mcr.microsoft.com/dynamicsnav:10.0.17763.1457-generic
96
97 Pulling generic image took 167 seconds
98
99
100 / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \
101 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
102 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
103 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
104 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
105 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
106 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
107
108 WARNING: Container name should not exceed 15 characters
109 BeContainerHelper is version 1.0.11
110 BeContainerHelper is running as administrator
111 Host is Microsoft Windows Server 2019 Datacenter - Itsc2019
112 Docker Client Version is 19.03.12
113 Docker Server Version is 19.03.12
114 Downloading application artifact /sandbox/17.0.17126.17791/us
115 Downloading C:\Users\VssAdministrator\AppData\Local\Temp\32076f5c-1535-471b-a7a0-4370af5ae286.zip

```

The Run Pipeline is the task, which basically performs the same steps as you did when running the local devenv pipeline earlier and you will see similar output.

```

231
232 Files:
233 http://hostedagent-myapp-ci:8080/Alllanguage.vsix
234
235 Container Total Physical Memory is 7.0Gb
236 Container Free Physical Memory is 2.7Gb
237
238 Initialization took 88 seconds
239 Ready for connections!
240 Reading CustomSettings.config from hostedagent-myapp-ci
241 Creating Desktop Shortcuts for hostedagent-myapp-ci
242 Container hostedagent-myapp-ci successfully created
243
244 Use:
245 Get-BcContainerEventLog -containerName hostedagent-myapp-ci to retrieve a snapshot of the event log from the container
246 Get-BcContainerDebugInfo -containerName hostedagent-myapp-ci to get debug information about the container
247 Enter-BcContainer -containerName hostedagent-myapp-ci to open a PowerShell prompt inside the container
248 Remove-BcContainer -containerName hostedagent-myapp-ci to remove the container again
249 docker logs hostedagent-myapp-ci to retrieve information about URL's again
250
251 Creating container took 759 seconds
252
253
254 / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \   / \
255 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
256 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
257 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
258 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
259 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
260 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
261
262 Using Version 1.0.2.0
263 Using Symbols Folder: D:\a\1\s\packages

```

The pipeline is by default setup to use Azure Hosted Agents, which will take approx. 20 minutes to complete a build and nothing can really be reused between builds.

You can also setup your own build agents (either on an Azure VM or a local computer) for use with the build, which will greatly increase speed but will also come with a cost of running and maintaining that machine.

When the build is complete, you should see:

The screenshot shows the Azure DevOps interface for a pipeline named "myfirstapp". The left sidebar is visible with various project management options like Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The "Pipelines" option is selected. The main content area displays the build log for run #20201025.2. The log title is "Build" with a green checkmark icon. The log details the following steps:

- 1 Pool: Azure Pipelines
- 2 Image: Windows-Latest
- 3 Agent: Hosted Agent
- 4 Started: Today at 4:27 PM
- 5 Duration: 20m 2s
- 6
- 7 Job preparation parameters
- 8 1 artifact produced
- 9 100% tests passed
- 10 Job live console data:
- 11 Starting: Build
- 12 Finishing: Build

Below these, a detailed list of individual job steps is shown:

- Initialize job 3s
- Pre-job: Download secr... 1s
- Download secrets: Bui... <1s
- Checkout myfirstapp@... 3s
- Set BuildNumber 2s
- Run Pipeline 19m 16s
- Publish Test Results 10s
- Publish Artifacts 1s
- Cleanup 20s
- Post-job: Checkout myfi... 1s
- Finalize Job <1s
- Report build status <1s

A search bar and several navigation icons are located at the top right of the main content area.

And clicking **view raw log** on the **Run Pipeline** will give you a nice output of the pipeline, which also can be used to search for into, and should always be included when creating issues on Run-Pipeline.

```
Pipelines - Run 20201025.2 logs x https://dev.azure.com/fkristiansen/7868ede4-2ac8-41a1-b7ee-1341447b8222/_apis/build/builds/2/logs/9
2020-10-25T15:45:22.692459Z D:\a\1\s\output\My Name\_My Test App_1.0.2.0.app successfully created in 13 seconds
2020-10-25T15:45:22.687817Z My Name\_My Test App_1.0.2.0.app copied to D:\a\1\s\packages
2020-10-25T15:45:22.710935Z
2020-10-25T15:45:22.710826Z Compiling apps, test apps and importing test toolkit took 105 seconds
2020-10-25T15:45:22.711084Z
2020-10-25T15:45:22.712587Z
2020-10-25T15:45:22.713084Z
2020-10-25T15:45:22.714365Z
2020-10-25T15:45:22.715920Z
2020-10-25T15:45:22.719398Z
2020-10-25T15:45:22.717836Z
2020-10-25T15:45:22.723434Z
2020-10-25T15:45:22.723870Z
2020-10-25T15:45:22.724071Z
2020-10-25T15:45:25.2089630Z Publishing c:\sources\output\My Name\_My Base App_1.0.2.0.app
2020-10-25T15:45:29.105704Z Synchronizing My Base App on tenant default
2020-10-25T15:45:29.2747050Z Installing My Base App on tenant default
2020-10-25T15:45:29.4792590Z App successfully published
2020-10-25T15:45:29.9045856Z Publishing c:\sources\output\My Name\_My App_1.0.2.0.app
2020-10-25T15:45:30.1856598Z Synchronizing My App on tenant default
2020-10-25T15:45:30.8226154Z Installing My App on tenant default
2020-10-25T15:45:31.0000000Z Publishing c:\sources\output\My Name\_My Test App_1.0.2.0.app
2020-10-25T15:45:31.5021752Z Publishing c:\sources\output\My Name\_My Test App_1.0.2.0.app
2020-10-25T15:45:31.6872678Z Synchronizing My Test App on tenant default
2020-10-25T15:45:31.8547653Z Installing My Test App on tenant default
2020-10-25T15:45:32.0594458Z App successfully published
2020-10-25T15:45:32.0512295Z
2020-10-25T15:45:32.0514742Z Publishing apps took 9 seconds
2020-10-25T15:45:32.0515412Z
2020-10-25T15:45:32.0655079Z
2020-10-25T15:45:32.0673499Z
2020-10-25T15:45:32.0677217Z
2020-10-25T15:45:32.0891790Z
2020-10-25T15:45:32.0938065Z
2020-10-25T15:45:32.0949864Z
2020-10-25T15:45:32.0981117Z
2020-10-25T15:45:32.1006676Z
2020-10-25T15:45:32.1010684Z
2020-10-25T15:46:48.7416161Z Codeunit 80133 HelloWorld Test Success (13.824 seconds)
2020-10-25T15:46:48.7589166Z Testfunction TestHelloWorldMessage Success (13.824 seconds)
2020-10-25T15:46:49.4873413Z
2020-10-25T15:46:49.4916035Z Running tests took 77 seconds
2020-10-25T15:46:49.4963359Z
2020-10-25T15:46:49.4980295Z
2020-10-25T15:46:49.5052794Z
2020-10-25T15:46:49.5060194Z
2020-10-25T15:46:49.5245215Z
2020-10-25T15:46:49.5272038Z
2020-10-25T15:46:49.5300391Z Getting Runtime Package For My Name\_My Base App_1.0.2.0.app
2020-10-25T15:46:50.7576588Z Copying runtime package to build artifact
2020-10-25T15:46:50.7701715Z Getting Runtime Package For My Name\_My App_1.0.2.0.app
2020-10-25T15:46:51.0976484Z Copying runtime package to build artifact
```

You can inspect the test results:

A screenshot of the Azure DevOps Pipelines test results page. The URL is [https://dev.azure.com/fkristiansen/myfirstapp/\\_build/results?buildId=2&view=ms.vss-test-web.build-test-results-tab](https://dev.azure.com/fkristiansen/myfirstapp/_build/results?buildId=2&view=ms.vss-test-web.build-test-results-tab). The pipeline name is "myfirstapp". The test run summary shows 1 run completed (1 passed, 0 failed), 100% pass percentage, and a run duration of 13s 823ms. A large trophy icon is displayed with the message "Hooray! There are no test failures."

And in the build summary, you can download the published app artifacts (not to be confused with Business Central artifacts):

A screenshot of the Azure DevOps Pipelines build summary page. The URL is [https://dev.azure.com/fkristiansen/myfirstapp/\\_build/results?buildId=2&view=results](https://dev.azure.com/fkristiansen/myfirstapp/_build/results?buildId=2&view=results). The pipeline name is "myfirstapp". The build summary shows it was manually run by Freddy Kristiansen. The repository and version is myfirstapp, master, e44e7d1. The time started and elapsed is Today at 4:24 PM, 20m 6s. Related work items: 0. Tests and coverage: 100% passed. Artifacts: 1 published. A table shows the build job status: Name (Build), Status (Success), Duration (20m 2s).

The app is available as app and runtime package. The test app and the test results are also published.

Note that the runtime packages are prefixed with a number series, which indicates the order in which they should be installed (dependencies first), as Sort-AppFilesByDependencies doesn't work on runtime packages.

The screenshot shows the Azure DevOps interface for a project named 'myfirstapp'. The left sidebar is visible with various navigation options like Overview, Boards, Repos, Pipelines, Test Plans, and Artifacts. The 'Pipelines' option is selected. The main content area is titled 'Artifacts' and shows a table of published artifacts for build 2. The table has two columns: 'Name' and 'Size'. The artifacts listed are:

Name	Size
output	27 KB
Apps	5 KB
My Name_My App_1.0.2.0.app	3 KB
My Name_My Base App_1.0.2.0.app	3 KB
RuntimePackages	8 KB
01 - My Name_My Base App_1.0.2.0.runtime.app	4 KB
02 - My Name_My App_1.0.2.0.runtime.app	5 KB
TestApps	3 KB
My Name_My Test App_1.0.2.0.app	3 KB
testresults.xml	12 KB

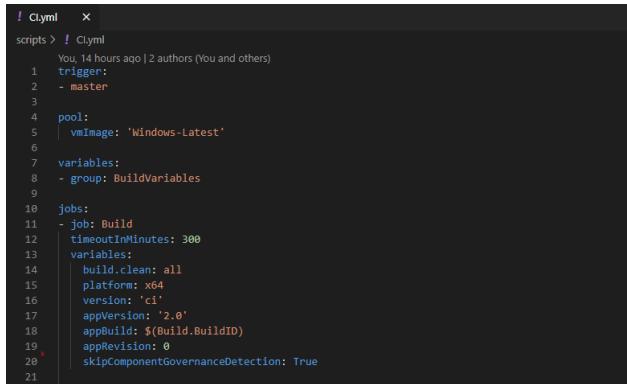
**Congratulations – you have run your first Build pipeline.**

# Inspect the pipeline

Let's have a look at the pipeline and some of the steps.

## Initialization

In CI.yml you have some settings and variables at the start:



```
! CI.yml ×
scripts > ! CI.yml
You: 14 hours ago | 2 authors (You and others)
1 trigger:
2 - master
3
4 pool:
5 | vImage: 'Windows-Latest'
6
7 variables:
8 - group: BuildVariables
9
10 jobs:
11 - job: Build
12   timeoutInMinutes: 300
13   variables:
14     build.clean: all
15     platform: x64
16     version: 'ci'
17     appVersion: '2.0'
18     appBuild: ${Build.BuildID}
19     appRevision: 0
20     skipComponentGovernanceDetection: True
21
```

**Trigger: master** means that the pipeline will trigger whenever a change is made to the master branch.

**Pool: vImage: 'windows-latest'** means that the pipeline will use Azure Hosted agents of the latest Windows version. Change this to **name: 'default'** if you want to use self-hosted agents in the default pool. (You can use <https://aka.ms/getbuildagent> to create a self-hosted agent and add it to the default pool).

**Variables: group: buildVariables** will request access to the variable group buildVariables and make the variables defined in that available in the pipeline.

**timeoutInMinutes: 300** determines that the build will run in up to 300 minutes before timing out.

**build.clean: all** means that DevOps will clean all symbols caches and binary folders before every build – do not attempt to reuse anything.

**platform: x64** indicates that we want to run 64 bit mode (not 32 bit x86) **pool: name: Default** says that we want to use a build agent from the Default pool (in which we places our agent)

**version: 'ci'** determines which settings from **settings.json** to use. The settings points out which Business Central artifacts to use for the build, which apps to install, the App Folders, the Test Folders, which test framework to install, which cops to enable etc. etc.

**AppVersion, AppBuild and AppRevision** determines the version numbering. There are a lot of different ways to make version numbering of your app. I have implemented one very simple model, where the three variables **appVersion**, **appBuild** and **appRevision** will be combined into a version number of the build. **appBuild** is set to the **Build\_BuildID** which is a unique build ID (auto incrementing) for this Organization.

The individual apps compiled by the pipeline will get the same **appBuild** and **appRevision**, but will keep their **appMajor** and **appMinor** from **app.json**. This allows for dependency apps to be compiled from various pipelines and get unique build numbers.

## Steps

### Set BuildNumber

Inline PowerShell script to set the build number for Azure DevOps.

### Run Pipeline

Invoke the function DevOps-Pipeline to perform the actual pipeline. InsiderSasToken, LicenseFile and code signing certificate are transferred from variable group to function in environment variables as the build agent won't have

access to key vaults, the pipeline does. AppBuild and AppRevision are transferred as parameters together with version, which indicates whether this should be a ci build, a current build, a nextminor build or a nextmajor build.

The DevOps-Pipeline will also check that the right version of BcContainerHelper is installed and imported and it will read the settings file to determine the right settings for invoking the Run-APipeline in BcContainerHelper

#### Publish Test Results

Publish the test results, which are saves in XML files by the pipeline. The format used for the test output is Junit.

#### Publish Artifacts

Publish the build artifacts (not to be confused with artifacts for running a container) to devops. This is the build result. The apps, the runtime packages, the test apps and the test results.

#### Cleanup

Cleanup the environment by invoking cleanup.ps1. When using hosted agents, this is really not necessary, they get cleaned up automatically. When using self-hosted agents, the cleanup function will remove containers, artifacts and images left over after failed builds. Artifacts and images, which haven't been used for 2 days are removed.

## Settings

The settings file in the scripts folder determines the settings for running the pipeline. The file is read and used only by the Read-Settings file, which reads the file and sets a number of variables based on settings.json. Read-Settings takes a version parameter indicating which version to build (ci, current, nextminor or nextmajor). Settings are read from the specific version section if available, else from the main section.

This means that if you want to run **AppSourceCop** in all versions but the **ci** version – you would set this to **true in the main section and false in the ci section**.

Read-Settings will set the following variables

**agentName** is blank if running local or set to \$ENV:AGENT\_NAME if running a devops agent.

**pipelineName** is the name of the pipeline, including the version (e.g. HelloWorld-ci)

**containerName** is set to the name of the build container. The agent name is included in the name if running a devops agent.

The following list are the settings, which will be turned into variables with the same name

**installApps** is a comma separated list of secure url's to dependencies (apps or .zip files containing apps) to be installed before compiling apps and test apps. The apps can be apps or runtime packages. If they are apps, they are sorted after dependencies before installing. Runtime packages are sorted and installed alphabetically.

**previousApps** is a comma separated list of secure url's to previous versions of the apps (or .zip files containing previous versions of the apps) to be used as previous versions of apps for **AppSourceCop** breaking change detection and upgrade test. When **previousApps** are specifying, these apps are published and installed before the newly build apps and upgrade is run before the tests.

**appFolders** is a comma separated list of folders which should be compiled as apps. The folders will be sorted after dependencies before compiled, published and installed. Apps in app folders are compiled before the test framework/libraries are published/installed and they are signed (if signing certificate is specified) unless doNotSignApp is set to true.

**appSourceCopMandatoryAffixes** is a comma separated list of affixes to be used as mandatory affixes in AppSourceCop settings.

**appSourceCopSupportedCountries** is a comma separated list of supported countries to be used as supported countries in AppSourceCop settings.

**testFolders** is a comma separated list of folders which should be compiled and used as test apps. The folders will be sorted after dependencies before compiled, published and installed. Apps in test folders are compiled after the test framework/libraries are published/installed and tests in these apps are used for test execution.

**memoryLimit** determines the amount of memory available in the build container during the pipeline run.

**additionalCountries** is a comma separated list of country codes. During the pipeline, a container with this local version of Business Central will be spun up and the apps produced will be published, installed and tested. Test results will be gathered for all versions. Default is no additional countries.

**genericImageName** is the generic image name to use for creating the container. Default is the default generic image in BcContainerHelper configuration.

**vaultNameForLocal** is the name of the key vault to use for secrets like licensefile, insiderSasToken and passwords. This setting is only used in **Local-DevEnv.ps1** and **Local-Pipeline.ps1**.

**bcContainerHelperVersion** determines which version of BcContainerHelper to use. Latest is the default setting, which probably is fine for most. Preview means grab the latest preview version and a specific version number will grab that exact version from the PowerShell Gallery. This setting can also be a local path on the build agent or a URL to a github repository, where the desired version of the containerhelper can be downloaded.

**installTestFramework** is a Boolean setting determining whether to install the Test Framework before compiling the test apps.

**installTestLibraries** is a Boolean setting, determining whether to install the Test Libraries before compiling the test apps. Test Libraries includes the Test Framework.

**installPerformanceToolkit** is a Boolean setting, determining whether to install the Performance Toolkit before compiling the test apps. Performance toolkit include the Test Framework.

**enableCodeCop** is a Boolean setting, determining whether Code Cop is enabled.

**enableAppSourceCop** is a Boolean setting, determining whether AppSource Cop is enabled.

**enablePerTenantExtensionCop** is a Boolean setting, determining whether Per Tenant Extension Cop is enabled.

**enableUICop** is a Boolean setting, determining whether UI Cop is enabled.

**doNotSignApps** is a Boolean setting which can be set to true if you do not want to sign Apps.

**doNotRunTests** is a Boolean setting, which can be set to true if you do not want to run Tests.

**cachelImage** is a Boolean setting, which determines whether an image will be cached before creating the build container. By default cachelImage is set to true on ci pipelines which is typically reusing the same Business Central version and false on other pipelines as versions changes a lot.

## Publishing Test Results

The **Publish Test Results** step will publish the **JUnit** compatible test results file to **Azure DevOps**, giving you the opportunity to investigate failing tests, see stack traces and creating work item for fixing failing tests.

A screenshot of the Azure DevOps Pipelines test results page. The URL is [https://dev.azure.com/fkristiansen/myfirstapp/\\_build/results?buildId=4&view=ms.vss-test-web.build-test-results-tab](https://dev.azure.com/fkristiansen/myfirstapp/_build/results?buildId=4&view=ms.vss-test-web.build-test-results-tab). The pipeline name is "myfirstapp". A green checkmark icon indicates a successful run titled "#2.0.4.0 fix cleanup". The summary shows 1 total test, 1 passed, 0 failed, and 0 others. The pass percentage is 100%. The run duration was 11s 490ms. A trophy icon is displayed with the message "Hooray! There are no test failures."

## Publish Artifacts

The **Publish Artifacts** step will publish the generated build artifacts to make them available for a release pipeline. The artifacts contains the .app files generated (including tests), runtime packages of the same apps and the test results.

The artifacts are easily consumed by release pipelines.

A screenshot of the Azure DevOps Pipelines Published artifacts page. The URL is [https://dev.azure.com/fkristiansen/myfirstapp/\\_build/results?buildId=4&view=artifacts&type=publishedArtifacts](https://dev.azure.com/fkristiansen/myfirstapp/_build/results?buildId=4&view=artifacts&type=publishedArtifacts). The pipeline name is "myfirstapp". The artifacts section shows a table with the following data:

Name	Size
output	27 KB
Apps	5 KB
My Name_My App_1.0.4.0.app	3 KB
My Name_My Base App_1.0.4.0.app	3 KB
RuntimePackages	8 KB
01 - My Name_My Base App_1.0.4.0.runtime.app	4 KB
02 - My Name_My App_1.0.4.0.runtime.app	5 KB
TestApps	3 KB
My Name_My Test App_1.0.4.0.app	3 KB
testresults.xml	12 KB

## Cleanup

The Cleanup step will invoke the **cleanup.ps1** script, which will remove the container (if present) and cleanup up artifacts cache and container images, which hasn't been used the last 2 days.

# Create a Release Pipeline

I will be working with a few different release pipelines.

1. Two release pipelines for releasing the build artifacts to blob storage (preview and production). You can also use Azure DevOps Artifacts, but what I really like about blob storage is, that I can secure access to the artifacts as I decide and I can download the artifacts using a simple Url (no PowerShell commands with special Az modules needed). The preview release pipeline should be invoked after every successful build, the production one should run on demand.
2. Two release pipelines for releasing a Per Tenant Extension to an online environment for a customer to test in sandbox environment or run in production. The pipeline releasing to a sandbox environment could be setup to run after every successful build, the one releasing to production should run on demand.
3. **Later versions** of this workshop will also include how to **deploy the App directly to AppSource** and through that **to end-customers** who have installed your app.

## Releasing to Blob Storage

The way I have structured the blob storage for my apps is:

<https://storageaccount/appname/version/apps.zip>

appname could be **bingmaps**, **bingmaps-preview**, **helloworld** or **helloworld-preview**. Version is either a **specific version number** or **latest**.

Examples:

<https://businesscentralapps.blob.core.windows.net/bingmaps/16.0.10208.0/apps.zip>

gives me version 16.0.10208.0 of the BingMaps app, and

<https://businesscentralapps.blob.core.windows.net/bingmaps-preview/latest/runtimepackages.zip>

gives me the runtime packages from the latest version of the bingmaps preview, and

<https://businesscentralapps.blob.core.windows.net/helloworld/latest/apps.zip>

gives me the latest production release of the helloworld app.

You might have noticed that I always use the latest production release of my apps as **previousapps** setting for AppSourceCop to get breaking change notification. So, let's setup the two release pipelines.

### Create a Blob Storage and get the Connection String

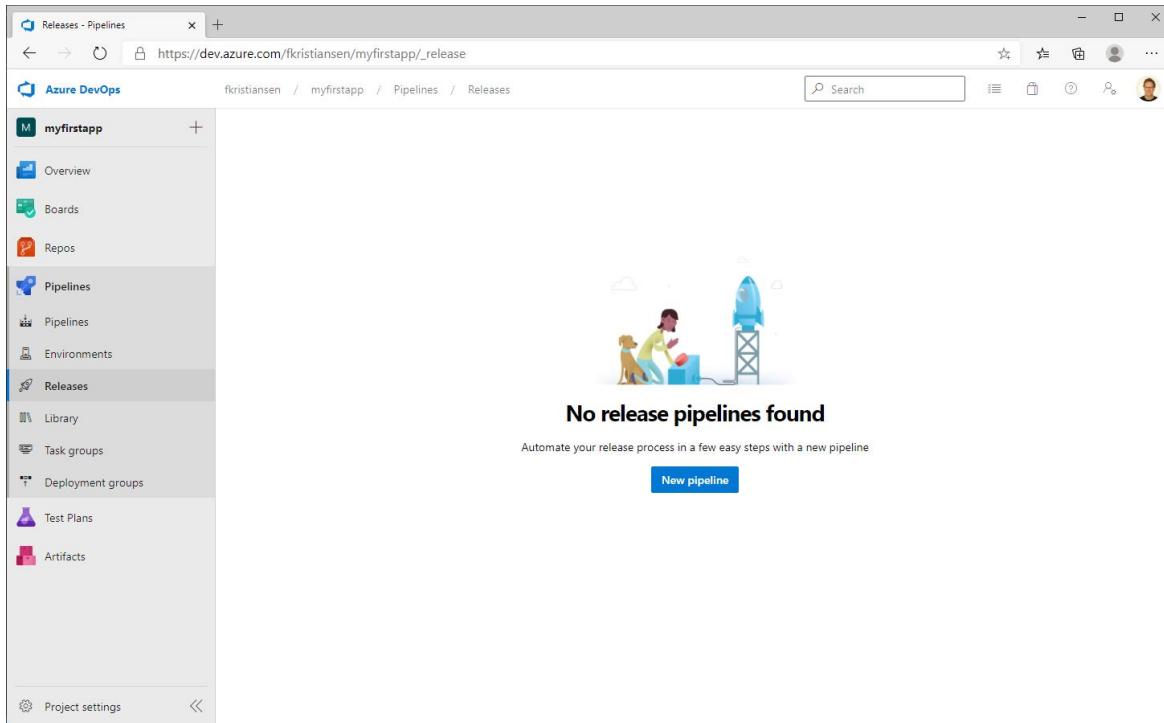
First we need a Blob Storage and a connection string. The Blog Storage Account is created in the Azure Portal and under shared access signature you define the access requirements for the connection string and generate the connection string:

After pressing **Generate SAS and connection string**, you get three values:

Click the **Copy to Clipboard** next to the **Connection String** and add this as a secret in your key vault from the first section. In the library section under Pipelines, add the new secret to the secrets available to the pipelines, press Ok and Save.

## Creating the release Pipeline

In Azure DevOps, under **Pipelines**, click **Releases** and select **New pipeline**.



## Select Empty job as template

A screenshot of the 'New release pipeline - Pipelines' page. The URL is https://dev.azure.com/fkristiansen/myfirstapp/\_releaseDefinition?definitionId=0&amp;a=action-create-definition&amp;source=ReleaseNew&amp;path=%5C. The left sidebar shows 'myfirstapp' with 'Releases' selected. The main area shows a 'Pipeline' configuration screen with sections for 'Artifacts' and 'Stages'. A callout box highlights the 'Stage 1 Select a template' section. To the right, a 'Select a template' pane is open, showing a search bar and a list of featured templates: 'Azure App Service deployment', 'Deploy a Java app to Azure App Service', 'Deploy a Node.js app to Azure App Service', 'Deploy a PHP app to Azure App Service and Azure Database for MySQL', 'Deploy a Python app to Azure App Service and Azure database for MySQL', 'Deploy to a Kubernetes cluster', and 'IIS website and SQL database deployment'. An 'Others' section is also visible at the bottom.

## Click Add an artifact

New release pipeline - Pipelines

All pipelines > New release pipeline

Stage Stage 1

Properties

Name and owners of the stage

Stage name Stage 1

Stage owner Freddy Kristiansen

Use the **build pipeline** as source and name the artifacts **Artifacts**.

Add an artifact

Source type

Build

5 more artifact types

Project

myfirstapp

Source (build pipeline)

myfirstapp

Default version

Latest

Source alias

Artifacts

The artifacts published by each version will be available for deployment in release pipelines. The latest successful build of myfirstapp published the following artifacts: output.

Add

Click Tasks and click the + next to the Agent Job. Search for PowerShell and add a PowerShell task to the job.

The **PowerShell Script task** displays that some settings needs attention. Click the **PowerShell Script task**. Select **Inline** and Paste the following code into the script editor:

```
Write-Host "Installing BcContainerHelper"
Install-Module BcContainerHelper -Force

Write-Host "Publishing to Storage"
Publish-BuildOutputToStorage
    -storageConnectionString "$(StorageConnectionString)"
    -projectId "$($Env:BUILD_PROJECTNAME)-preview".ToLowerInvariant()
    -appVersion "$Env:BUILD_BUILDNUMBER"
    -path "Artifacts\output"
    -setLatest
```

Click **Variables**, **Variable Groups**, select **BuildVariables** and click **Link** to link the variable group to the release pipeline.

Set the name of the release Pipeline to **Release to Storage (preview)**. Click the small lightning bolt icon in artifacts to enable Continuous deployment trigger on the preview release pipeline.

Click **Save**. After successfully saving the release pipeline, click **Create Release** to create a release and you should get your artifacts published to blob storage:

Now redo the entire process again, where you remove **-preview** in the script and replace **(preview)** with **(prod)** in the name and I can now use <https://businesscentralapps.blob.core.windows.net/myfirstapp/latest/apps.zip> as previousapps in my settings file.

## Releasing a Per Tenant Extension to an online environment

To release per tenant extension apps to an online environment, we will use the service-to-service authentication for automation APIs, which was shipped in Business Central 2020 release wave 2 (v17). AJ did a very detailed description of this feature on his blog: <https://www.kauffmann.nl/2020/09/14/service-to-service-authentication-for-automation-apis-in-business-central/>.

The partner creates an AAD Application for authentication. The Customer registers the partners AAD Application in their Business Central tenant and assigns permissions that the app can do automation and extension management.

So, we need to setup a few things before we can create a release pipeline:

- An AAD Application for authenticating to Business Central
- An online tenant of Business Central v17 or higher
- Permissions in Business Central to allow the AAD App to do Automation and Extension Management

### An AAD Application for authenticating to Business Central

Every partner will probably create just one multitenant AAD Application, which they will use for managing extensions or their customers, so I am not going to create a script for this (not at this time 😊)

Open the Azure Portal (<https://portal.azure.com>), search for **App Registrations**, create a **New App Registration**. Give your app a **friendly display name**, set the app to **multitenant**, set the **redirect URI** to <https://businesscentral.dynamics.com/OAuthLanding.htm>. Click Register.

**Name:** Freddys App Publisher App

**Supported account types:** Accounts in any organizational directory (Any Azure AD directory - Multitenant)

**Redirect URI (optional):** https://businesscentral.dynamics.com/OAuthLanding.htm

Make a copy the Application (Client ID).

**Application (client) ID:** a26651f5-0e90-473c-b4f9-e96119aac8b8

**Documentation:**

- Microsoft identity platform
- Authentication scenarios
- Authentication libraries
- Code samples
- Microsoft Graph
- Glossary
- Help and Support

Click API Permissions and select + Add a permission, locate Dynamics 365 Business Central and select that.

The screenshot shows the Microsoft Azure portal's 'Request API permissions' interface. On the left, there's a sidebar with various service icons. The main area shows a list of pre-defined API permissions, each with a title, description, and a small icon. One item, 'Dynamics 365 Business Central', is highlighted with an orange border. At the bottom right of the main area, there are two buttons: 'Add permission' and 'Discard'.

Select Application Permissions, check Automation.ReadWrite.All and click Add permissions

This screenshot shows the 'Request API permissions' screen again, but with a different focus. The 'Select permissions' section on the right is expanded, showing a list of permissions under 'Automation (1)'. The 'Automation.ReadWrite.All' permission is checked and highlighted with a gray background. Below this list are two buttons: 'Add permissions' and 'Discard'.

Select Certificates & secrets, click + New client secret, give it a name, set the expiration date and click Add

The screenshot shows the 'Certificates & secrets' blade for an Azure app registration. A new client secret is being added with the description 'App Publisher Client Secret' and 'Never' selected for expiration. The 'Client secrets' table is empty.

Make a Copy of the Client Secret, it is only shown once

The screenshot shows the 'Certificates & secrets' blade for an Azure app registration. The 'Client secrets' table now shows a single row for 'App Publisher Client Secret' with a value copied from the previous screen. A note says 'Copy the new client secret value. You won't be able to retrieve it after you perform another operation or leave this blade.'

## Online tenant of Business Central

Next thing we need is an online tenant of Business Central, and we need to assign permissions to the partner AAD Application. In my admin center, I have created a production environment (MyProd) and a Sandbox environment (Sandbox), which I will use for this workshop.

The screenshot shows the Dynamics 365 Business Central admin center interface. On the left, there's a navigation sidebar with sections like 'Notification Recipients', 'Telemetry', 'Reported Outages', 'Operations', and 'Capacity'. The main area is titled 'Environments' and displays a table of environments. The columns are 'Name', 'Application Family', 'Type', 'State', 'Country/region', 'Current Version', 'Available Update Version', and 'Scheduled Up'. Three environments are listed: 'Production' (Business Central, Production, Active, US, 17.0.16993.0), 'MyProd' (Business Central, Production, Active, US, 17.0.16993.0), and 'Sandbox' (Business Central, Sandbox, Active, US, 17.0.16993.0). There are buttons for '+ New' and 'Refresh' at the top of the table.

Permissions in Business Central to allow the AAD App to do Automation and Extension Management  
Navigate to your **Business Central environment**, click search and enter **aad app**

The screenshot shows the Dynamics 365 Business Central home page for 'CRONUS USA, Inc.'. The page features a headline 'Good morning, Kristiansen!', a sales summary for the month, and an 'Ongoing Sales' dashboard. A search bar at the top has 'aad app' typed into it. A search results overlay is displayed, showing sections for 'Go to Pages and Tasks' (with 'AAD Applications' selected), 'Documentation' (with links to 'Transfer Data from a QuickBooks App', 'Use Your Data to Create an App', and 'Using Business Central Apps in Power BI'), and 'Get from Microsoft AppSource' (with links to 'IN Apps Analyzer', 'AR Collections App', and 'AR Electronic Funds Transfer App'). The overlay also includes a message ' Didn't find what you were looking for? Try exploring'.

Select AAD Applications, click + New, enter the Client ID and a description and click Grant Consent

The screenshot shows the Dynamics 365 Business Central interface. On the left, there's a navigation bar with various icons and links. The main area is titled "AAD Application Card" and shows a card for "Freddys App Publisher App". The card has sections for "General", "User Groups", and "User Permission Sets". Under "General", the Client Id is listed as {a26651f5-0e90-473c-b4f9-e9611}, the Description is "Freddys App Publisher App", and the State is "Enabled". The "User Groups" section shows a table with one row: CRONUS USA, Inc. under "Name". The "User Permission Sets" section is currently empty. At the top of the card, there's a reminder: "Reminder: your work date is 4/6/2020 Use today | Change to... | Turn off reminder".

You will now be asked to authenticate and give access to Freddys App Publisher App to do automation and sign in and read user profile

The screenshot shows a Microsoft Edge browser window with the title "Sign in to your account - BC SaaS - Microsoft Edge". The URL is https://login.microsoftonline.com/common/reprocess?ctx=rQIAeNiUGC0UnE0MjMzNX... The page displays a "Permissions requested" dialog from "Freddys App Publisher App (unverified)". It asks for "Full access to automation" and "Sign in and read user profile". It also states that accepting these permissions allows the app to access resources for all users in the organization. There is a note about the publisher not providing terms of service or privacy statements. At the bottom, there are "Cancel" and "Accept" buttons.

**Click Accept, and assign two User Permission Sets: D365 AUTOMATION and D365 EXTENSION MGT**

The screenshot shows the Dynamics 365 Business Central Admin Center interface. A modal window titled "New - AAD Application Card" is open. In the "Grant Consent" section, the Client ID is listed as `1e20001112-0e50-473c-0413-e901133d0001`, the Description is "Freddys App Publisher App", the State is "Enabled", and the App Name is listed as `[00000000-0000-0000-0000-000000000000]`. Below this, the "User Groups" section shows a table with one row assigned to the company "CRONUS USA, Inc.". Under "User Permission Sets", there are two rows: "D365 AUTOMATION" and "D365 EXTENSION MGT".

**Perform the same steps in the Sandbox environment.**

### Creating the release Pipeline

Open the Azure Portal and locate your Key Vault. Add secrets for **PublisherAppClientId** (Client ID from your AAD App) and **PublisherAppClientSecret** (Client Secret from your AAD App).

In Azure DevOps, navigate to Pipelines -> Library, modify build variables and add **PublisherAppClientId** and **PublisherAppClientSecret** to get access to these values for the pipelines and Save.

The screenshot shows the Azure DevOps Library page for the project "myfirstapp". On the left, the navigation menu is visible with options like Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, Artifacts, and Compliance. The "Library" tab is selected. In the main area, under "BuildVariables", there is a table of variables:

Delete	Secret name	Content type	Status	Expiration date
	InsiderSasToken		Enabled	3/15/2021 12:00:00 AM
	Licensefile		Enabled	Never
	Password		Enabled	Never
	StorageConne...		Enabled	Never
	PublisherApp...		Enabled	Never
	PublisherApp...		Enabled	Never

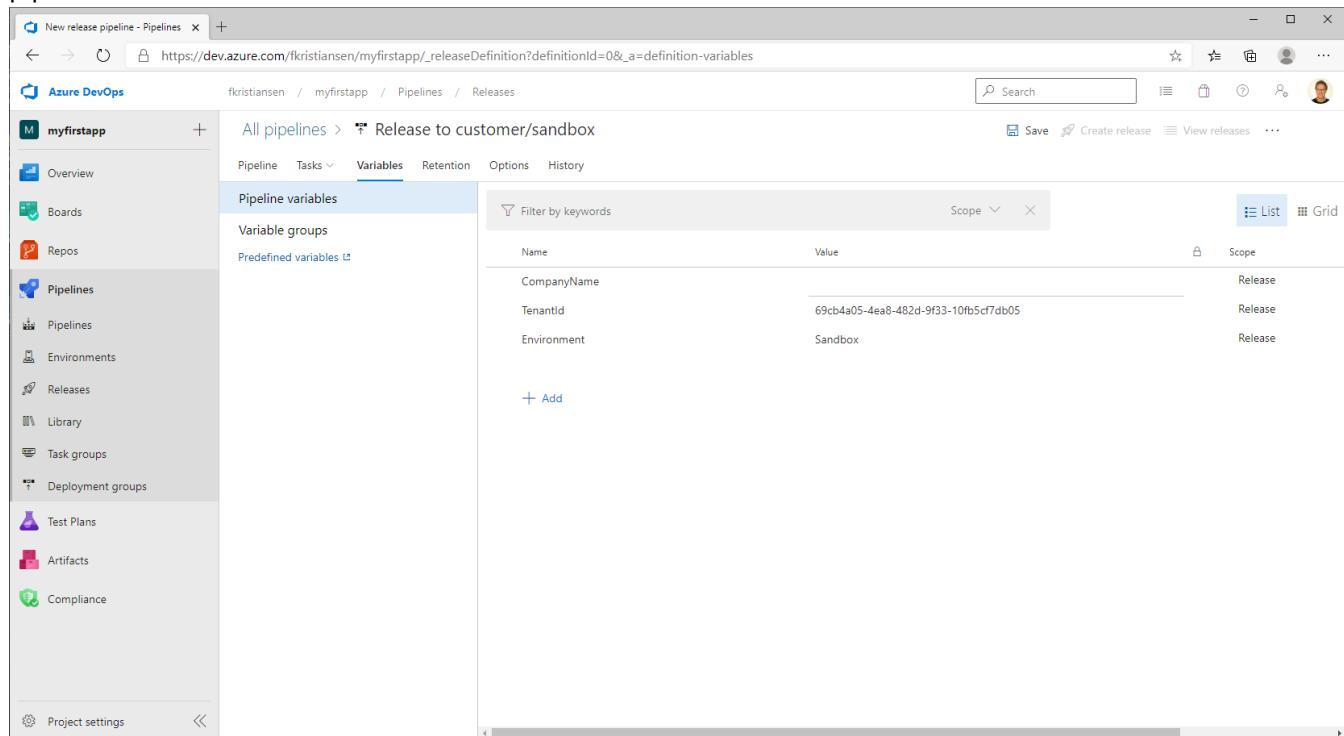
In Releases, click + New, add a new Release Pipeline, click Add an artifact, select the source and set the source alias to Artifacts, click Add.

The screenshot shows the 'New release pipeline' page in Azure DevOps. On the left, the navigation bar includes 'Overview', 'Boards', 'Repos', 'Pipelines', 'Environments', 'Releases', 'Library', 'Task groups', 'Deployment groups', 'Test Plans', 'Artifacts', and 'Compliance'. The 'Pipelines' item is selected. The main area shows 'All pipelines > New release pipeline'. On the left, there are sections for 'Artifacts' and 'Stages'. The 'Artifacts' section has a button 'Add an artifact'. The 'Stages' section shows 'Stage 1' with '1 job, 0 task'. On the right, the 'Add an artifact' dialog is open. Under 'Source type', 'Build' is selected. In the 'Project' dropdown, 'myfirstapp' is chosen. In the 'Source (build pipeline)' dropdown, 'myfirstapp' is also chosen. In the 'Default version' dropdown, 'Latest' is selected. In the 'Source alias' input field, 'Artifacts' is typed. A note at the bottom says: 'The artifacts published by each version will be available for deployment in release pipelines. The latest successful build of myfirstapp published the following artifacts: output.' A large blue 'Add' button is at the bottom right of the dialog.

In Variables, Variable Groups, link the variable group BuildVariables

The screenshot shows the 'Variables' tab in the 'New release pipeline' interface. The left sidebar is identical to the previous screenshot. The 'Variables' tab is selected. On the left, there are sections for 'Pipeline variables', 'Variable groups', and 'Predefined variables'. A 'Link variable group' button is visible. On the right, a modal dialog titled 'Link variable group' is open. It shows a search bar and a list containing 'BuildVariables (6)'. Below the list, 'Variable group scope' is set to 'Release'. A 'Select' dropdown is shown. At the bottom is a large blue 'Link' button.

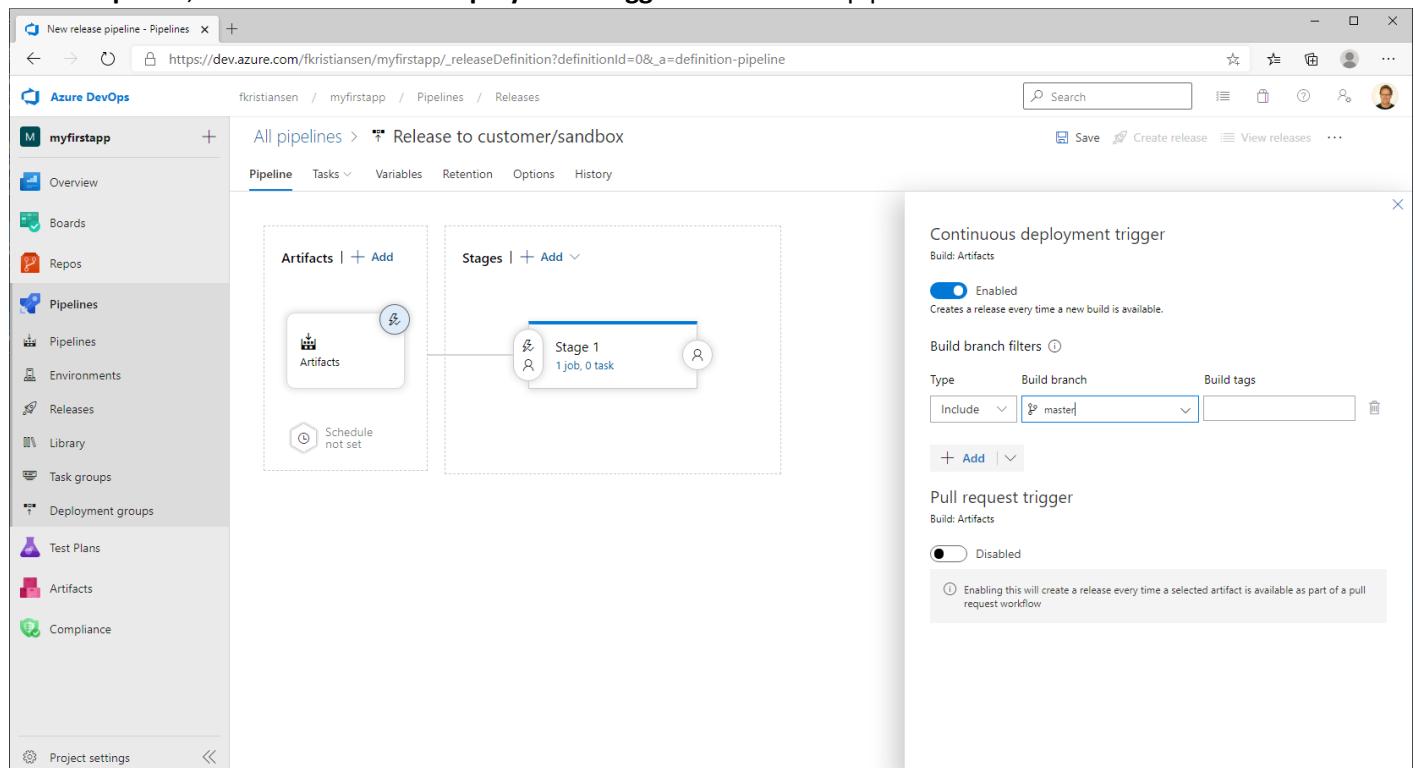
Under pipeline variables, define 3 variables: **CompanyName**, **TenantId** and **Environment**. Set the name of the pipeline to **Release to customer Sandbox**.



The screenshot shows the 'Pipeline variables' section of the Azure DevOps interface. It lists three predefined variables:

Name	Value	Scope
CompanyName	69cb4a05-4ea8-482d-9f33-10fb5cf7db05	Release
TenantId	Release	Release
Environment	Sandbox	Release

Under **Pipeline**, set the **Continuous Deployment Trigger** and **Save** the pipeline.



The screenshot shows the 'Pipeline' configuration screen. On the left, under 'Stages', there is one stage named 'Stage 1' which contains one job and no tasks. On the right, the 'Continuous deployment trigger' panel is open, showing the following settings:

- Build: Artifacts**: Enabled. Creates a release every time a new build is available.
- Build branch filters**: Type: Include, Build branch: master.
- Pull request trigger**: Build: Artifacts, Disabled. Enabling this will create a release every time a selected artifact is available as part of a pull request workflow.

Select Tasks, click Agent Job and select Windows-2019 as Agent Specification.

The screenshot shows the Azure DevOps Pipeline Editor. On the left, the navigation bar includes 'myfirstapp', 'Overview', 'Boards', 'Repos', 'Pipelines' (selected), 'Environments', 'Releases', 'Library', 'Task groups', 'Deployment groups', 'Test Plans', 'Artifacts', and 'Compliance'. The main area shows a pipeline named 'Release to customer/sandbox' with a single stage 'Stage 1: Deployment process'. Under 'Stage 1', there is an 'Agent job' section with a 'Run on agent' button. To the right, the 'Agent job' configuration pane is open, showing 'Agent job' details, 'Agent selection' (set to 'Azure Pipelines'), 'Agent Specification' (set to 'windows-2019'), 'Demands' (empty), 'Execution plan' (empty), 'Parallelism' (set to 'None'), and a 'Timeout' field.

Click the + in Agent job to add a job, search for PowerShell and add a PowerShell Task. Click the task line and set the type to be inline and copy/paste the following script:

```
Write-Host "Installing BcContainerHelper"
Install-Module BcContainerHelper -Force -AllowPrerelease

Write-Host "Publishing to tenant"
Publish-PerTenantExtensionApps -useNewLine
  -ClientID "$([System.Guid]::NewGuid())"
  -ClientSecret "$([System.Guid]::NewGuid())"
  -tenantId "$(TenantId)"
  -environment "$(Environment)"
  -companyName "$(CompanyName)"
  -appFiles @((Get-Item "Artifacts/output/Apps/*.app" | % { $_.FullName }))
```

The screenshot shows the Azure DevOps Pipeline Editor with the same pipeline structure as the previous screenshot. The 'Agent job' section now contains a 'PowerShell Script' task. The 'PowerShell Script' configuration pane is open, showing 'Task version' (set to '2.\*'), 'Display name' (set to 'PowerShell Script'), 'Type' (set to 'Inline'), and a 'Script' editor containing the PowerShell script from the previous step. Other sections like 'ErrorActionPreference', 'Advanced', 'Control Options', and 'Environment Variables' are also visible.

Save the pipeline and press **Create Release** to publish the latest version of your app to the **Sandbox** environment.

The screenshot shows the Azure DevOps Pipelines interface for a project named "myfirstapp". A new release named "Release-1" has been created. The "Tasks" tab is selected, showing a single "PowerShell Script" task under "Stage 1". The task is configured with the following script:

```
Write-Host "Installing BcContainerHelper"
Install-Module BcContainerHelper -Force -AllowPrerelease

Write-Host "Publishing to tenant"
Publish-PerTenantExtensionApps -useNewLine
-ClientId "$PublisherAppClientId"
-ClientSecret "$PublisherAppClientSecret"
-tenantId "$tenantId"
-environment "$Environment"
-companyName "$CompanyName"
```

The "Type" is set to "Inline". The "Script" field contains the PowerShell code above. The "ErrorActionPreference" is set to "Stop".

Click **Release-1** to monitor the pipeline

The screenshot shows the Azure DevOps Releases interface for the "Release-1" pipeline. The "Pipeline" tab is selected. The "Release" section shows it was manually triggered by "Freddy Kristiansen" on "10/27/2020, 8:22 AM". The "Stages" section shows "Stage 1" is "In progress" with 25 tasks pre-job download. Artifacts for version 2.0.10.0 from the master branch are listed.

## Running PowerShell Script

The screenshot shows the Azure DevOps Pipeline interface for a release named "Release to customer/sandbox". The pipeline has one stage, "Stage 1", which is currently "In progress". This stage contains a single task, "PowerShell Script". The logs for this task show the following output:

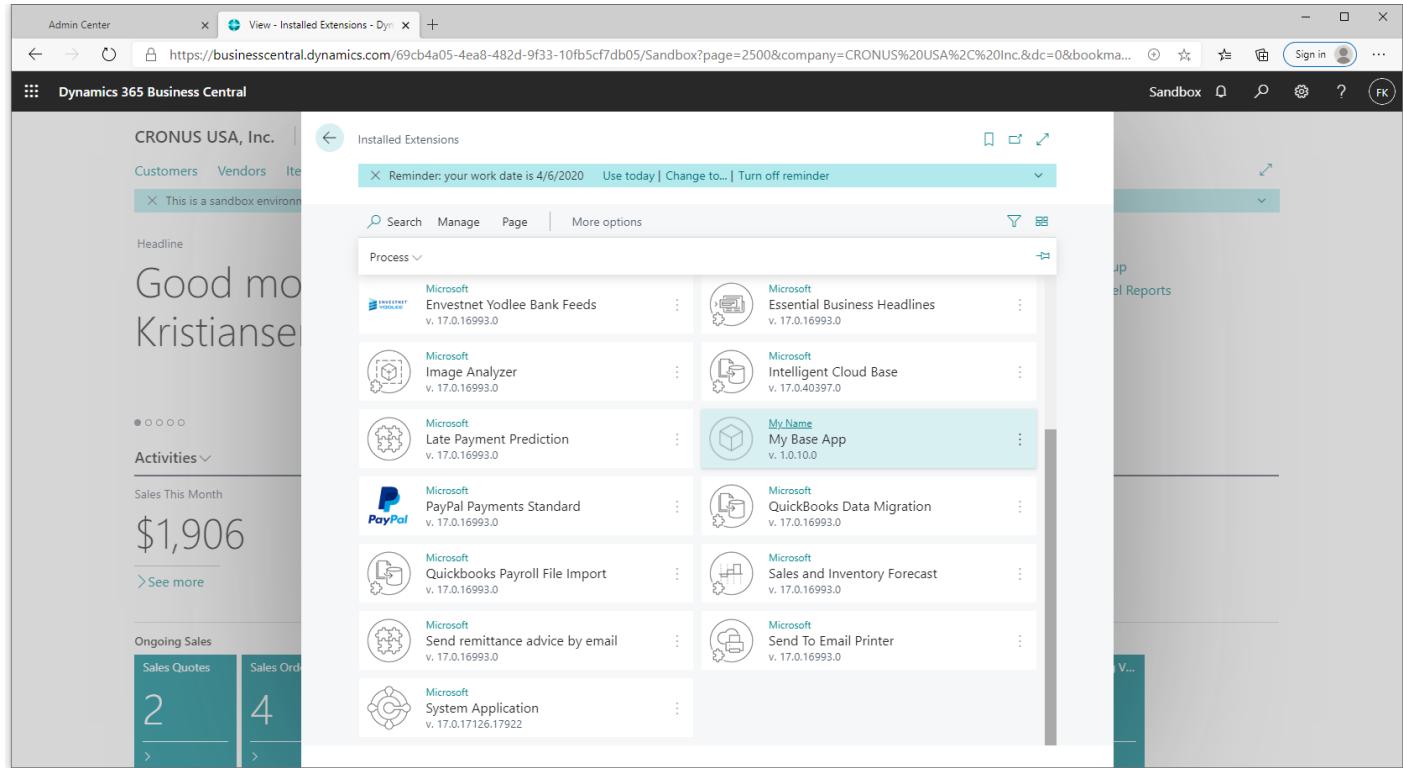
```
5/5 tasks completed in 00:23
PowerShell Script
[...]
```

Click Logs below Stage 1 to monitor the log

The screenshot shows the logs for Stage 1 of the pipeline. The logs are organized into sections: Deployment process, Agent job, and PowerShell Script. The PowerShell Script section contains the following log entries:

```
Initialize job - succeeded
Pre-job: Download secrets: BuildVariables skipped
Download artifact - Artifacts - output - succeeded
Download secrets: BuildVariables - succeeded
PowerShell Script
- Base Application, Version 17.0.17126.18816, Installed=True
- Business Central Cloud Migration - Previous Release, Version 17.0.40397.0, Installed=True
- Business Central Cloud Migration - Previous Release (US), Version 17.0.40397.0, Installed=True
- Business Central Intelligent Cloud, Version 17.0.40397.0, Installed=True
- Ceridian Payroll, Version 17.0.16993.0, Installed=True
- Company Hub, Version 17.0.16993.0, Installed=True
- D101 - Localization for Mexico, Version 17.0.16993.0, Installed=True
- Dynamics GP Intelligent Cloud, Version 17.0.40397.0, Installed=True
- Envestnet Yodlee Bank Feeds, Version 17.0.16993.0, Installed=True
- Essential Business Headlines, Version 17.0.16993.0, Installed=True
- Image Analyzer, Version 17.0.16993.0, Installed=True
- Intelligent Cloud Base, Version 17.0.40397.0, Installed=True
- Late Payment Prediction, Version 17.0.16993.0, Installed=True
- PayPal Payments Standard, Version 17.0.16993.0, Installed=True
- QuickBooks Data Migration, Version 17.0.16993.0, Installed=True
- Quickbooks Payroll File Import, Version 17.0.16993.0, Installed=True
- Sales and Inventory Forecast, Version 17.0.16993.0, Installed=True
- Send Remittance advice by email, Version 17.0.16993.0, Installed=True
- Send To Email Printer, Version 17.0.16993.0, Installed=True
- System Application, Version 17.0.17126.17922, Installed=True
My Name_My Base App_1.0.10.0.app
Publishing and Installing
[...]
```

You can also login to your **sandbox environment** and see that the apps are getting installed



You can repeat the process for the production tenant, without the continuous deployment trigger and deploy the app on demand.

**Congratulations, you have successfully published your app to an online tenant.**

## Branch Policies

At this time, we have an Azure DevOps project with a build pipeline, which we can kick off manually. We also have a release pipeline, which helps us release the product, but in order to raise the quality bar even more, we should setup branch policies to demand code reviews and successful builds before a fix makes it into the repository.

Open your Azure DevOps organization (eg. <https://dev.azure.com/>) and select **Repos** -> **Branches**. Select the **master** branch, click the **more actions** symbol (...) and choose **Branch Policies**.

The screenshot shows the 'Branches' page in Azure DevOps. On the left, there's a sidebar with options like Overview, Boards, Repos, Pipelines, etc. The 'Branches' section is selected. In the main area, the 'master' branch is listed with a commit by 'Fred...' from 'Yesterday'. A context menu is open over the 'master' branch row, with 'Branch policies' highlighted.

You should now be able to setup branch policies for your branch

The screenshot shows the 'Settings - Repositories' page for the 'myfirstapp' repository. Under 'Project Settings', the 'General' section is selected. In the main area, the 'master' branch is selected. The 'Branch Policies' section is expanded, showing four policy types: 'Require a minimum number of reviewers', 'Check for linked work items', 'Check for comment resolution', and 'Limit merge types'. Below these are sections for 'Build Validation' and 'Status Checks', both of which are currently empty.

The first thing you will notice is, that setting **any Required policy** will enforce the use of **pull requests** and will **disallow direct checkins** to the **master** branch. It will also prevent the branch from unintended or evil deletion.

You can read much more about the policies here: <https://docs.microsoft.com/en-us/azure/devops/repos/git/branch-policies>.

What I want, is to ensure that my CI build pipeline runs whenever somebody checks something in and that the checkin cannot be completed if the build isn't successful. This is called **Build validation** and by adding a build policy

with our CI pipeline, we should be ready to go.

The screenshot shows the Azure DevOps interface. On the left, there's a sidebar with 'Project Settings' for 'myfirstapp'. The main area shows 'Branch Policies' for the 'master' branch. It lists four policies: 'Require a minimum number of reviewers', 'Check for linked work items', 'Check for comment resolution', and 'Limit merge types'. Below these are sections for 'Build Validation' and 'Status Checks', both of which currently have zero items. To the right, a modal window titled 'Add build policy' is open. It has fields for 'Build pipeline \*' (set to 'myfirstapp'), 'Path filter (optional)', 'Trigger' (set to 'Automatic (whenever the source branch is updated)'), 'Policy requirement' (set to 'Required'), 'Build expiration' (set to 'After 12 hours if master has been updated'), and a 'Display name' field. At the bottom of the modal are 'Save' and 'Cancel' buttons.

## Congratulations

I know you think you are just getting started here,  
but with this, you have successfully completed this  
Hands On Lab.