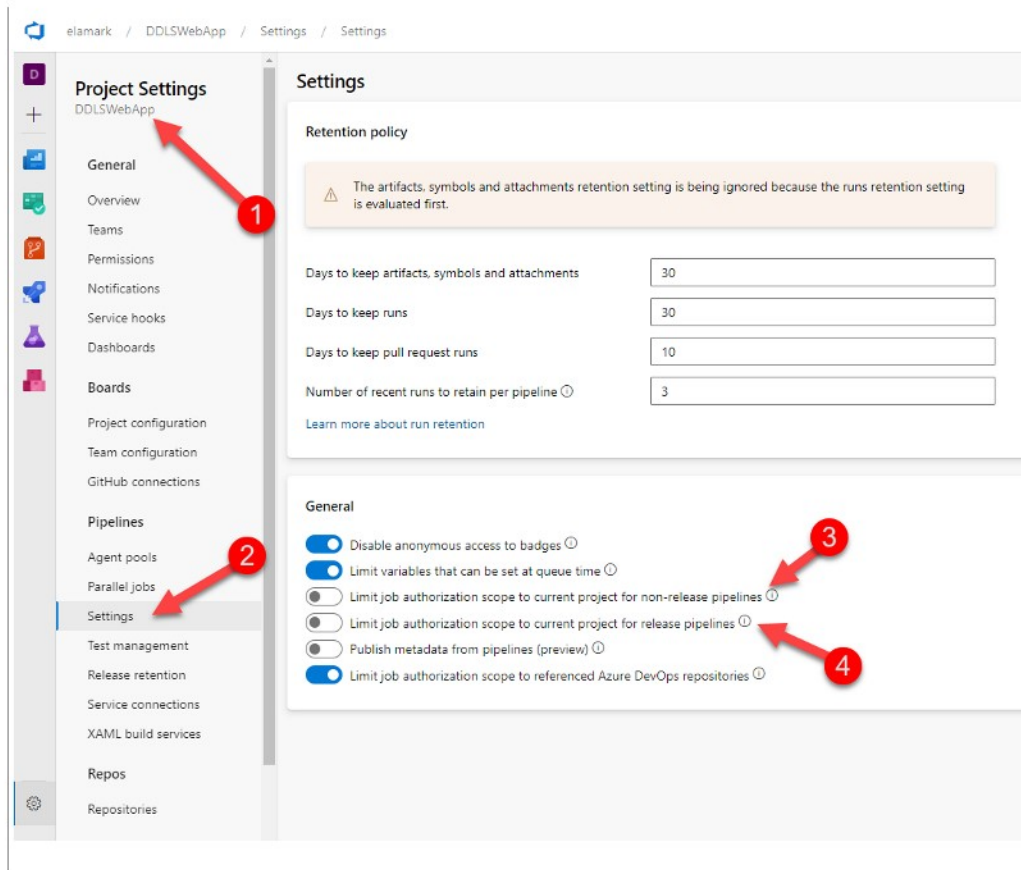


Release Pipeline demo

Thursday, 22 July 2021 3:02 PM

1. Create **two App Services in Azure** (**mwaz400webapp** and **mwaz400webapp-prod**). These will be the deployment targets
2. Create a **new project in DevOps** (**name=DDLWebApp**)
3. **Change the project settings as follows** (if you don't then the release pipeline will get a 401 when trying to access the DDLSUtilities artifact).

This demo depends on the DDLSUtilities Demo



4. **Initialise the Repo and clone to local machine.**
5. **dotnet new MVC**

```
public IActionResult Index()
{
    ViewBag.Result=DDLSUtilities.Class1.Cube(3);
    return View();
}
```

4. Initialise the Repo and clone to local machine.

5. **dotnet new MVC**

a. Change the application files:

i. **HomeController.cs**

ii. **index.cshtml**

```
{  
    ViewBag.Result=DDLSUtilities.Class1.Cube(3);  
    return View();  
}
```

6. **dotnet add package newtonsoft.json** (just to show we can use an upstream package)

7. Go to the **DDLSUtilities DevOps** project

a. -> **artifacts**

i. **connect to stream**

ii. **Copy the nuget.config file and add to the new MVC application**

```
@{  
    ViewData["Title"] = "Demo App";  
}  
  
<div class="text-center">  
    <h1 class="display-4">The Result=@ViewBag.Result !!!</h1>  
</div>
```

8. Back to the **MVC application**

a. **dotnet restore**

b. **dotnet run**

c. **git stage changes**

d. **git commit**

e. **git push**

1. Open the **DDLSWebApp project in DevOps**

a. Create a **new build pipeline** like this:

```
1 trigger:  
2  
3 --main  
4  
5  
6 pool:  
7 --vmImage: ubuntu-latest  
8  
9 steps:  
10  
11 Settings  
12 --task: DotNetCoreCLI@2  
13 --inputs:  
14 --command: 'restore'  
15 --projects: '**/*.csproj'  
16 --feedsToUse: 'select'  
17 --vstsFeed: '61f3e0e6-4949-4115-a054-00a7badac984/982c68bf-9b86-46ec-88a5-5590f5b9422b'  
18 --noCache: true  
19  
20 Settings  
21 --task: DotNetCoreCLI@2  
22 --inputs:  
23 --command: 'build'  
24 --projects: '**/*.csproj'
```

Choose this from the dropdown

Feeds and authentication

Feeds to use * ⓘ

☒ Feed(s) I select here

☐ Feeds in my NuGet.config

Use packages from this Azure Artifacts feed ⓘ

DDLSUtilities

☒ Use packages from NuGet.org ⓘ

```
$(Build.SourcesDirectory)  
$(Build.ArtifactStagingDirectory)
```

```

21 |..command: 'build'
22 |..projects: '**/*.csproj'
23
24 |Settings
25 |- task: DotNetCoreCLI@2
26 |..inputs:
27 |..command: 'publish'
28 |..publishWebProjects: true
29
30 |Settings
31 |- task: CopyFiles@2
32 |..inputs:
33 |..sourceFolder: '$(Build.SourcesDirectory)'
34 |..targetFolder: '$(Build.ArtifactStagingDirectory)'
35
36 |Settings
37 |- task: PublishBuildArtifacts@1
38 |..inputs:
39 |..PathToPublish: '$(Build.ArtifactStagingDirectory)'
40 |..ArtifactName: 'drop'
41 |..publishLocation: 'Container'

```

\$(Build.SourcesDirectory)
\$(Build.ArtifactStagingDirectory)

This is necessary or you'll get:

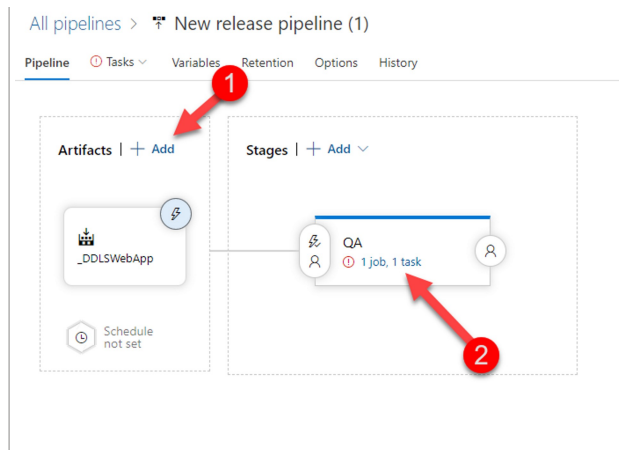
Warnings 1

Directory '/home/vsts/work/1/a' is empty. Nothing will be added to build artifact 'drop'.
PublishBuildArtifacts

12. run the build pipeline (should work)

13. Click on releases in DevOps

- a. Create a new release pipeline
 - i. select AzureAppService deployment
 - ii. change the name stage1 to QA
 - iii. click +Add (artifact) (1, below diagram)
 - 1) sourcetype=build
 - 2) source=DDLSWebApp
 - iv. Click to change tasks (2, below diagram)



v. Complete the Job as follows:

All pipelines > New release pipeline

Save Create release View releases

Pipeline Tasks Variables Retention Options History

QA-Web-App

Some settings need attention

Run on agent

Run on agent

Deploy Azure App Service

Azure App Service deploy

Stage name

QA-Web-App

Parameters

Unlink all

Azure subscription

Visual Studio Enterprise Subscription (2b6924e4-58ec-4492-8994-a152a4487152)

Scoped to subscription 'Visual Studio Enterprise Subscription'

This field is linked to 1 setting in 'Deploy Azure App Service'

App type

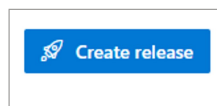
Web App on Windows

App service name

mwaz400webapp

This field is linked to 1 setting in 'Deploy Azure App Service'

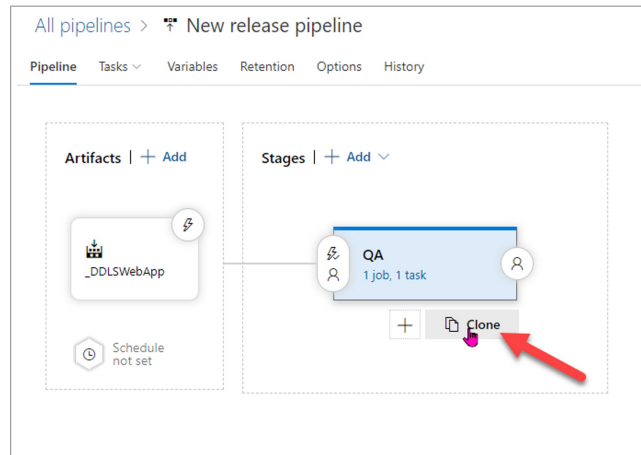
vi. Once the release pipeline has been created, click **create release**.



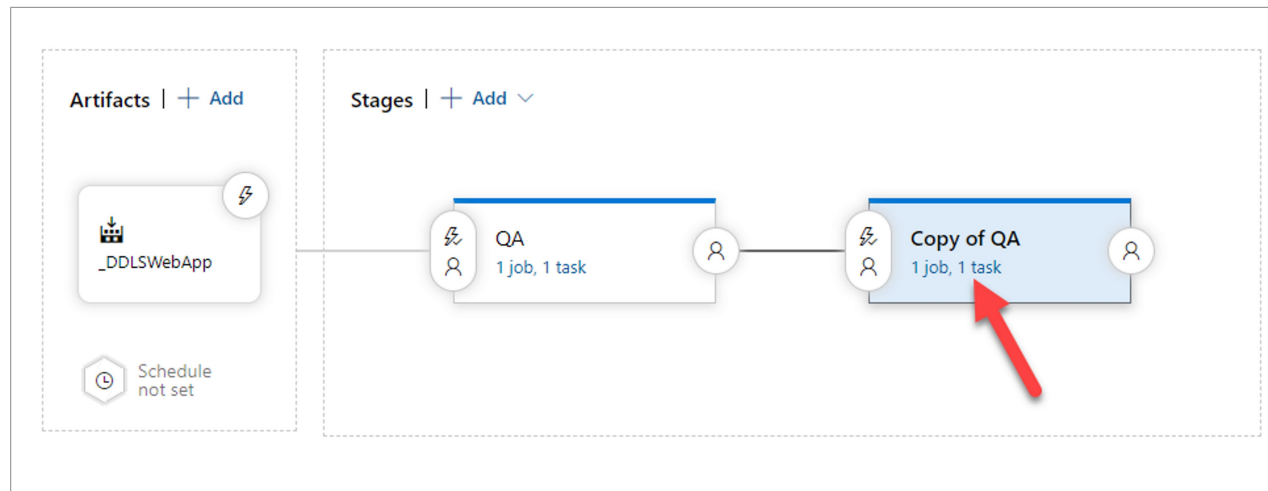
vii. Once the release is ready **click deploy**.

Next: Deploy to multiple Environments:

a. **Clone the QA Stage:**



b. **Change the target environment**



c. Get it to **point to the other Web App**

Stage name
Prod

Parameters ⓘ | Unlink all

Azure subscription * | Manage

Visual Studio Enterprise Subscription (2b6924e4-58ec-4492-8994-a152a4487152)

Scoped to subscription 'Visual Studio Enterprise Subscription'

App type

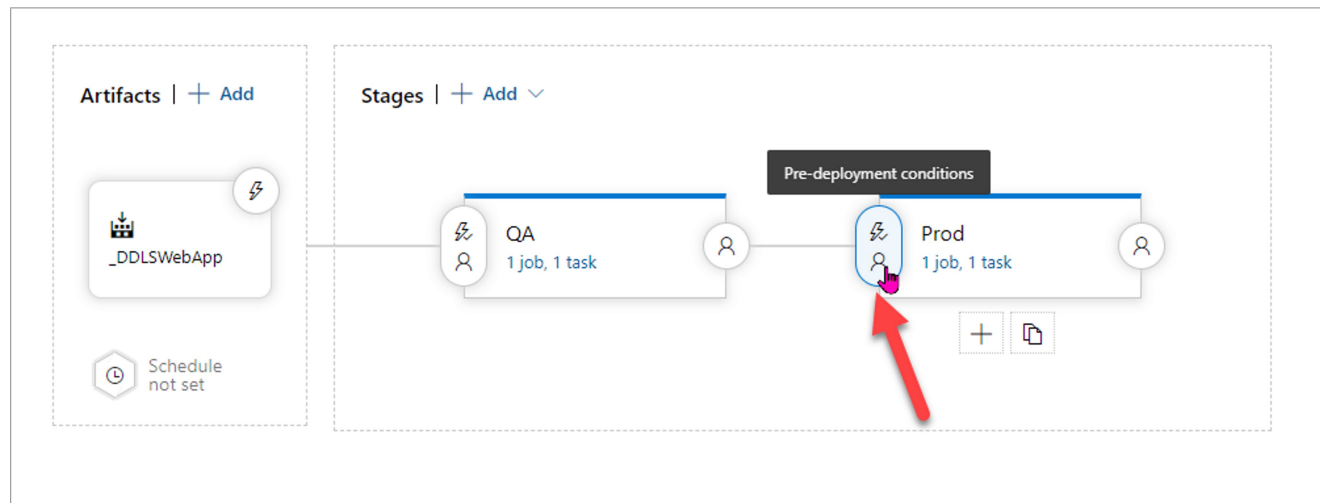
Web App on Windows

App service name * |

mwaz400webapp-prod

d. **Save, Create release, Deploy**

e. Go back and edit the release pipeline. **Add an approver to the PROD stage**



Pre-deployment conditions

Prod

Triggers ^

Define the trigger that will start deployment to this stage

Select trigger ⓘ

After release After stage Manual only

Stages ⓘ

QA

☐ Trigger even when the selected stages partially succeed ⓘ

Artifact filters ⓘ ☐ Disabled

Schedule ⓘ ☐ Disabled

Pull request deployment ⓘ ☐ Disabled

Pre-deployment approvals

Select the users who can approve or reject deployments to this stage

2

Pre-deployment conditions

Prod

Triggers v

Define the trigger that will start deployment to this stage

Pre-deployment approvals ^ ☒ Enabled

Select the users who can approve or reject deployments to this stage

Approvers ⓘ

Mark Walsh X Search users and groups for approvers

Timeout ⓘ

30 Days

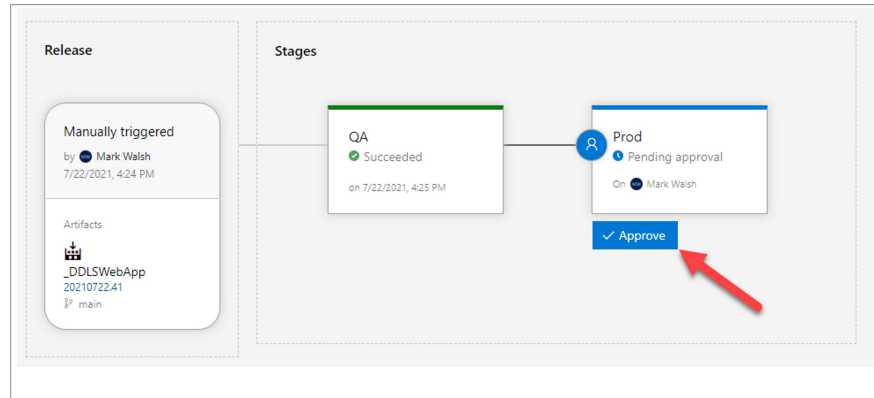
Approval policies

☐ The user requesting a release or deployment should not approve it

☐ Skip approval if the same approver approved the previous stage ⓘ

f. **Save, Create Release, Deploy**

g. **Press Approve.**



Could also try creating another release pipeline to deploy to a web app in the USA

