

Step By Step: Setup Continuous Delivery

Create Service Endpoint

To deploy our application towards Azure we need to have a service endpoint to be able to connect to Azure from VSTS. In your project there is a Team Administrator available. This account can be used to register the service endpoint. The following steps guide you through the process.

Create service principal

1. Verify if your machine is Azure PowerShell ready by running the following command
2. You are good if you see similar output

```
Directory: C:\Program Files\WindowsPowerShell\Modules

ModuleType Version      Name                               ExportedCommands
-----
Script      5.7.0      AzureRM
```

1. If not, please follow instructions here to configure this: <https://docs.microsoft.com/en-us/powershell/azure/install-azurerm-ps?view=azurerm-ps-5.7.0>
2. Download and run this PowerShell script in an Azure PowerShell window <https://github.com/Microsoft/vsts-rm-extensions/blob/master/TaskModules/powershell/Azure/SPNCreation.ps1>. Keep the values listed at the bottom available as well as the password you have chosen.

```
Windows PowerShell

D:\WORK\GDBC> .\SPNCreation.ps1

cmdlet azure.ps1 at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
SubscriptionName: Visual Studio Enterprise with MSDN
password: *****
Provide your credentials to access Azure subscription Visual Studio Enterprise with MSDN

Account      : jgilhuis@xpirit.com
SubscriptionName : Visual Studio Enterprise with MSDN
SubscriptionId : d0ceec8-6096-4a0b-afa4-
TenantId     : 6d959f2c-bd8b-44ab-a278-
Environment  : AzureCloud

WARNING: Unable to acquire token for tenant 'af84181f-1a46-441a-900f-3'
Creating a new Application in AAD (App URI - http://VSTS.xpiri.4258ce07-b724-41c2-
VERBOSE: Performing the operation 'Adding a new application with display name
'VSTS.xpiri.4258ce07-b724-41c2-8233-a7b4f62f9821' on target 'VSTS.xpiri.4258ce07-b724-41c2-8233-
Azure AAD Application creation completed successfully (Application Id: b182536f-b48f-48bc-a908-7bee)
Creating a new SPN
SPN creation completed successfully (SPN Name: )
Waiting for SPN creation to reflect in Directory before Role assignment
Assigning role (owner) to SPN App (b182536f-b48f-48bc-a908-7bee)

RoleAssignmentId : /subscriptions/d0ceec8-6096-4a0b-afa4-a5e0239bb198/providers/Microsoft.Authorization/roleAssignme
nts/6119e16e-d8df-4f7a-b981-283f4cc4632d
```

```

Scope       : /subscriptions/d0ceec8-6096-4a0b-af18-2b1a5b4e624
DisplayName  : VSTS.xpiri.4258ce07-b724-41c2-8f11-2b1a5b4e624
SignInName   :
RoleDefinitionName : Owner
RoleDefinitionId  : 8e3af657-a8ff-443c-a75c-2fe8c
ObjectId      : 562f1a5b-ae62-4377-ad12-ca3c4
ObjectType     : ServicePrincipal
CanDelegate    : False

SPN role assignment completed successfully

Copy and Paste below values for Service Connection
*****
Connection Name: (SPN)
Environment: AzureCloud
Subscription Id: d0ceec8-6096-4a0b-af18-2b1a5b4e624
Subscription Name:
Service Principal Id: b182536f-b48f-4811-80c3-2b1a5b4e624
Service Principal key: <Password that you typed in>
Tenant Id: 6d959f2c-bd8b-44ab-b118-2b1a5b4e624
*****

```

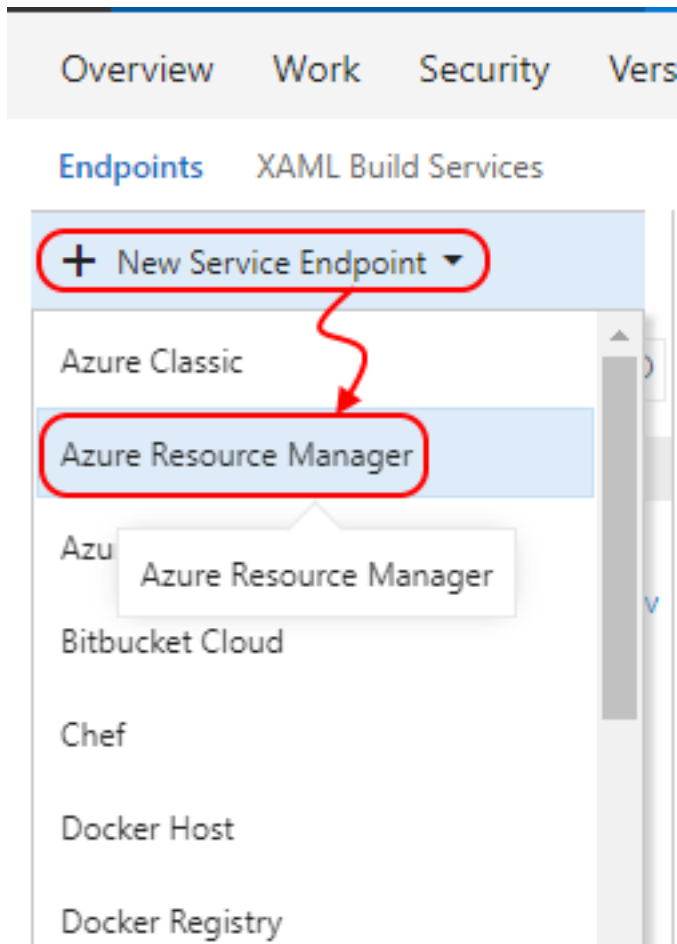
Note: some values have been modified in this screen, like subscription name

Setup Permissions

1. Navigate to the Project Admin Page in VSTS by clicking on the gear icon.
2. Click on the Security tab
3. Navigate to the Endpoint Creators Group
4. Add your Team or Teams' users to the Group

Create Service endpoint

1. Navigate to the Services Admin Page in VSTS by clicking on the gear icon.
2. Click on the Services tab
3. Choose to create a "New Service Endpoint" of type "Azure Resource Manager"



External Git

Generic

GitHub

1. From the dialogue choose to use the full version of the dialogue by clicking the highlighted link

×

Add Azure Resource Manager Service Endpoint

Connection name

Subscription

()

▼

Resource Group

▼

Subscriptions listed are from Azure Cloud

A new Azure Service Principal will be created and assigned with "Contributor" role, having access to all the resources in the selected subscription. If a Resource group is selected, access is restricted to all the resources within just the resource group.

If your subscription is not listed above, or your account is not backed by Azure Active Directory or to specify an existing Service Principal, [use the full version of the endpoint dialog.](#)

OK

Close

1. Fill in the dialogue

×

Add Azure Resource Manager Service Endpoint

Connection name

Environment

AzureCloud

▼

Subscription ID

Subscription Name

Service Principal Client ID

Service Principal Key

Tenant ID

Connection: Not verified

Verify connection


For help on creating an Azure Service Principal, see [Service endpoints](#).

To create a new Service Principal automatically, [use the automated version of the endpoint dialog](#).

OK

Close

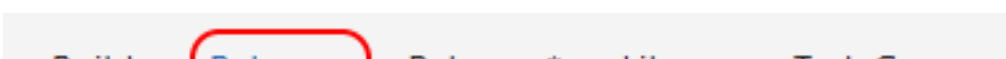
1. Enter the desired connection name e.g. **GDBC Team X**
2. Choose **"Azure Cloud"** as your environment
3. Enter your **Subscription ID**, copy from the script results
4. Enter your **Subscription Name**, copy from the script results
5. Enter your **Service Principal Client ID**, copy from the script results
6. Enter your **Service Principal Key**, copy from the script results
7. Enter your **Tenant ID**, copy from the script results
8. Verify your connection by clicking the "Verify Connection" link

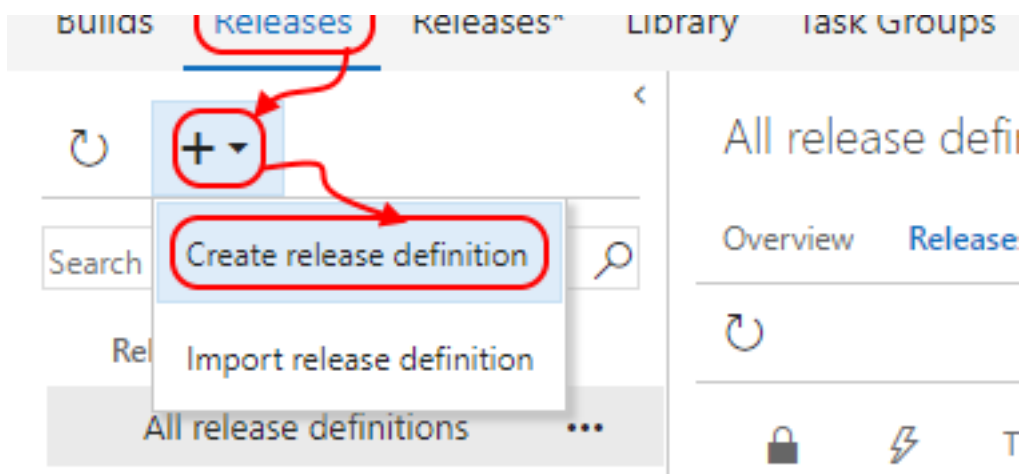
Connection:  Verified

Verify connection

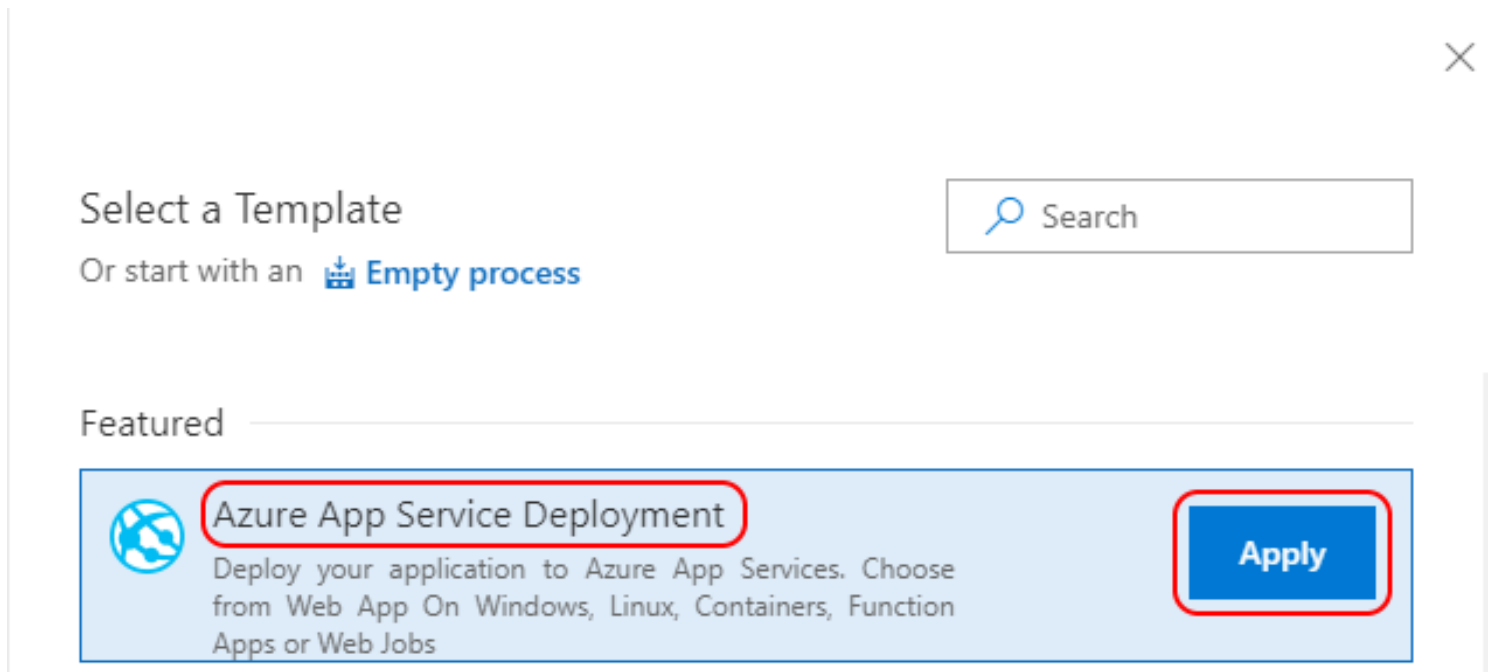
Create a Release Definition

1. Navigate to VSTS and select your project <https://globaldevopsbootcamp.visualstudio.com>
2. Navigate to the "Build and Release" hub
3. Create a new release definition

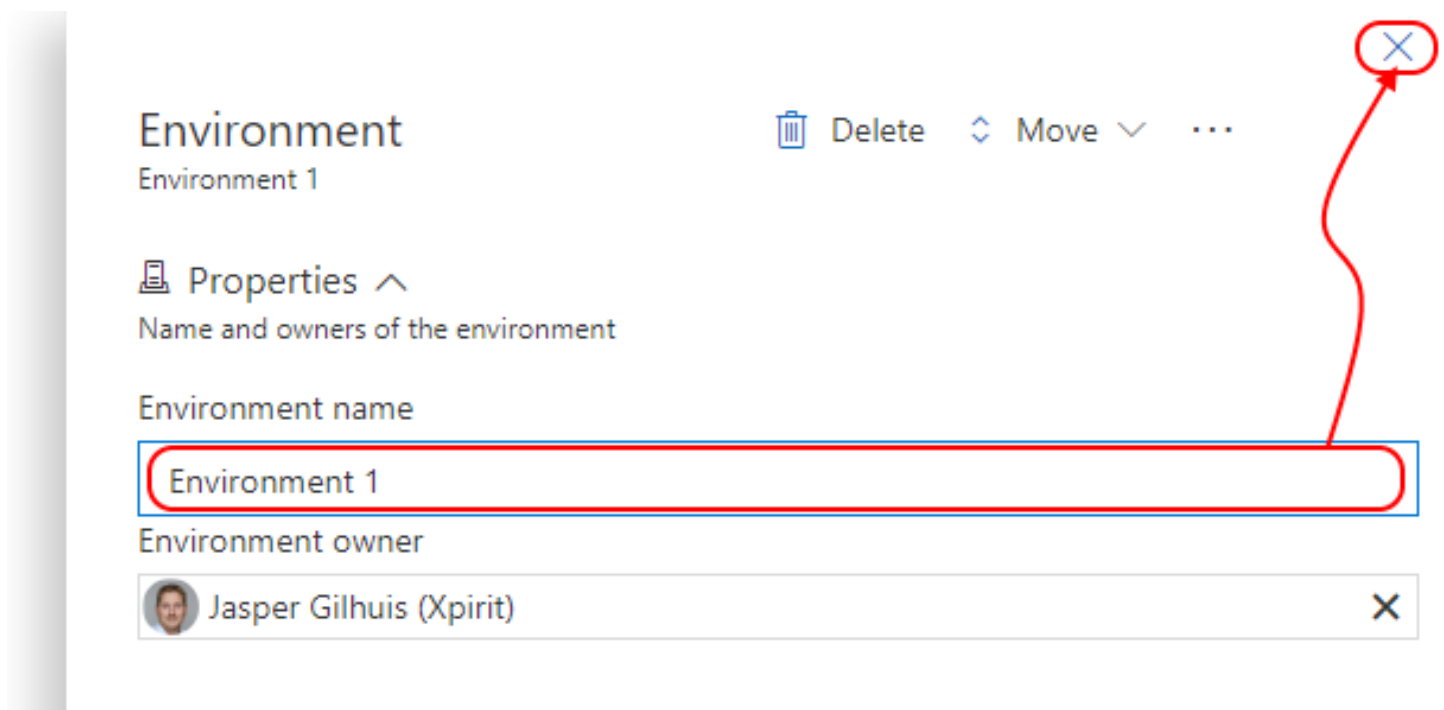




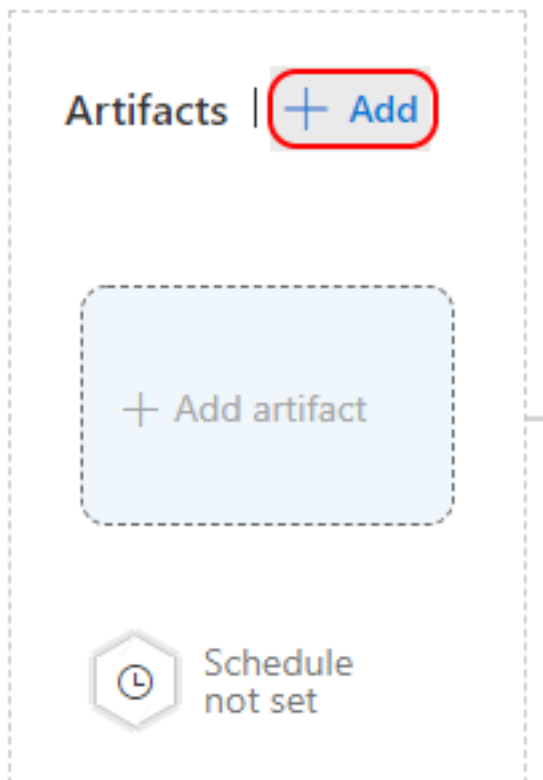
1. Choose the "Azure App Service Deployment" template



1. Provide a logical name to the environment, e.g. "Release Verification"



1. Add an artifact to the release



1. Select the source type "Build", the project should be pre-selected, choose the "Source" (Build Definition), and select "Latest" as version. Then provide a logical name, to be able to refer to the Artifact published by the build later. Then choose "Add" to add this artifact.

A screenshot of a dialog box titled 'Add artifact' with a close button (X) in the top right corner. Under the heading 'Source type', there are four icons: a factory icon with a checkmark and the text 'Build' (highlighted with a red box), a Git icon, a GitHub icon, and a Team Foundation icon. Below these is a link '4 more artifact types' with a downward arrow. The 'Project *' field is a dropdown menu with 'Scrum' selected and an information icon (i). The 'Source (Build definition) *' field is a dropdown menu with 'Sample Web Application' selected and an information icon (i). Red arrows point from the 'Build' icon to the 'Project' dropdown and from the 'Sample Web Application' dropdown to the 'Source' dropdown.

Default version * ⓘ

Latest

Source alias ⓘ

_Application

ⓘ The artifacts published by each version will be available for deployment in Release Management. The latest successful build of **Sample Web Application** published the following artifacts: **drop**, **build.SourceLabel**.

Add

1. Add another artifact to the release, this time choose "Git" as Source Type and then select the correct Source (Repository) as well as the Default Branch and Default version. Then provide a logical name to be able to refer to the Artifact to use in the release later.

Add artifact

Source type



Build



✓ Git



GitHub



Team Found...

4 more artifact types ▾

Project * ⓘ

Scrum

Source (repository) * ⓘ

DevOpsResources

Default branch * ⓘ

master

Default version * ⓘ

Latest from default branch

☐ Checkout submodules ⓘ

☐ Checkout files from LFS ⓘ

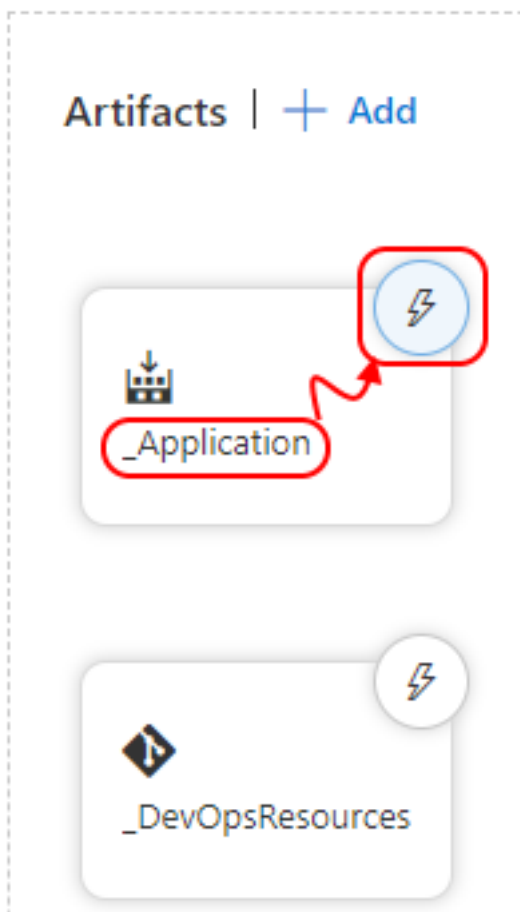
Shallow fetch depth ⓘ

Source alias ⓘ

_DevOpsResources

Add


1. For the 'Application artifact' set the "Continuous deployment trigger"



1. Enable the Continuous deployment trigger

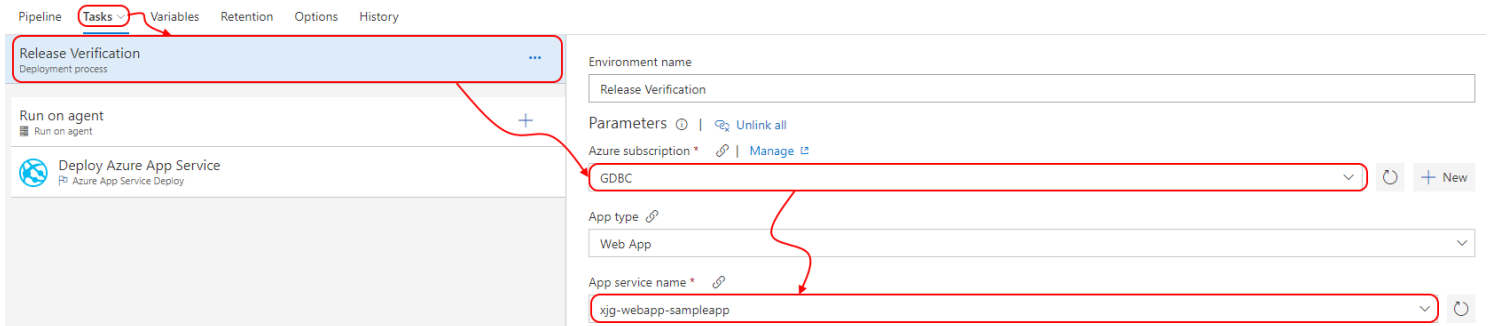
Continuous deployment trigger

Build: _Application

 Enabled

Creates release every time a new build is available.

1. Navigate to the "Tasks" tab
2. On your the environment section, choose your azure subscription, and provide a "App Service Name".
Choose to use a variable for this e.g. **xjg-webapp-sampleapp**.



Pipeline **Tasks** Variables Retention Options History

Release Verification
Deployment process

Run on agent
Run on agent

Deploy Azure App Service
Azure App Service Deployment

Environment name
Release Verification

Parameters | Unlink all

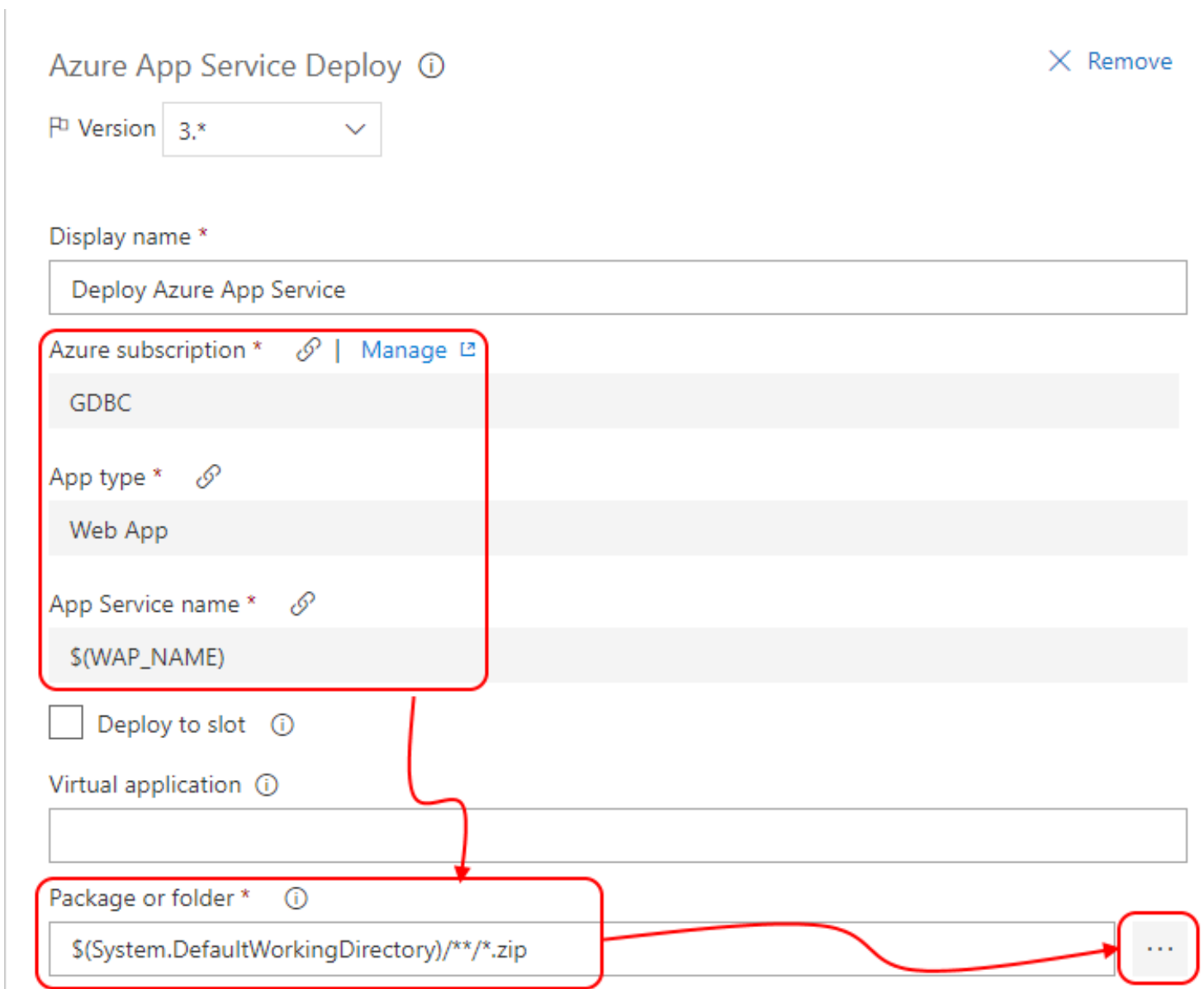
Azure subscription * | Manage
GDBC

App type
Web App

App service name *
xjg-webapp-sampleapp

Note: The name of your app service name should be globally unique

1. Now select the Azure App Service Deployment task, Verify the details. It should match previous selected items. Notice the Package selection. Click the "..." button to select the our application zipfile.



Azure App Service Deploy ⓘ Remove

Version 3.*

Display name *
Deploy Azure App Service

Azure subscription * | Manage
GDBC

App type *
Web App

App Service name *
\$(WAP_NAME)

☐ Deploy to slot ⓘ

Virtual application ⓘ

Package or folder * ⓘ
\$(System.DefaultWorkingDirectory)/**/*.zip

...

File Transforms & Variable Substitution Options ▾

Additional Deployment Options ▾

Post Deployment Action ▾

Application and Configuration Settings ▾

Output ^

App Service URL ⓘ

Control Options ▾

Output Variables ▾

1. Browse to the correct artifact (Build) and select the zip file of our application

×

Select File Or Folder

Linked Artifacts

_Application (Build)

drop

Sample Web Application.deploy-readme.txt

Sample Web Application.deploy.cmd

Sample Web Application.SetParameters.xml

Sample Web Application.SourceManifest.xml

Sample Web Application.zip

build.SourceLabel

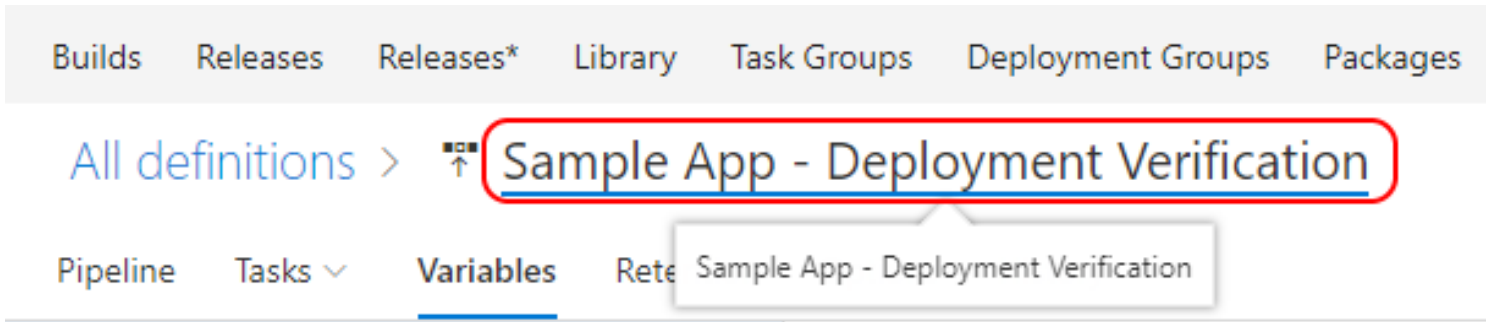
_DevOpsResources (Git)

The artifacts published by each version will be available for deployment in Release Management. The last successful version of `_Application (Build)` published the following artifacts: `drop`, `build.SourceLabel`.

Location `_Application/drop/Sample Web Application.zip`



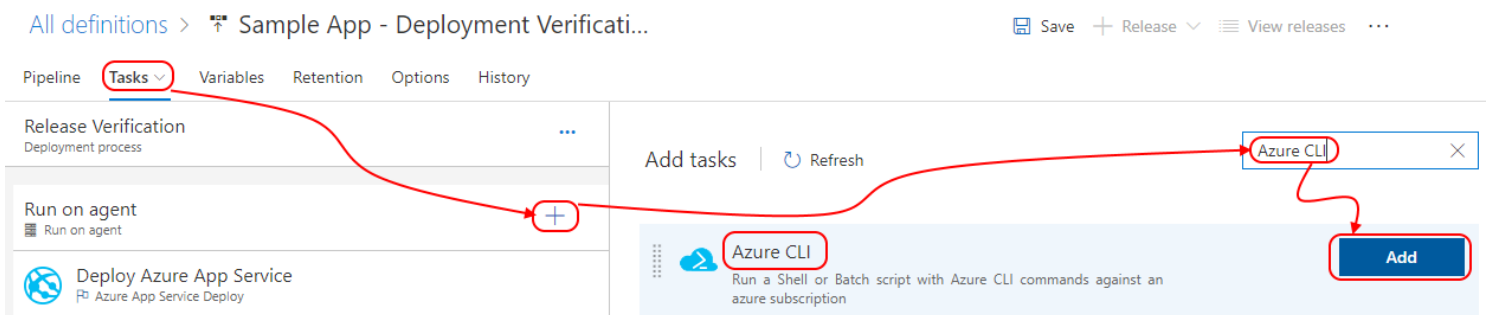
1. Rename your release, and "Save" it



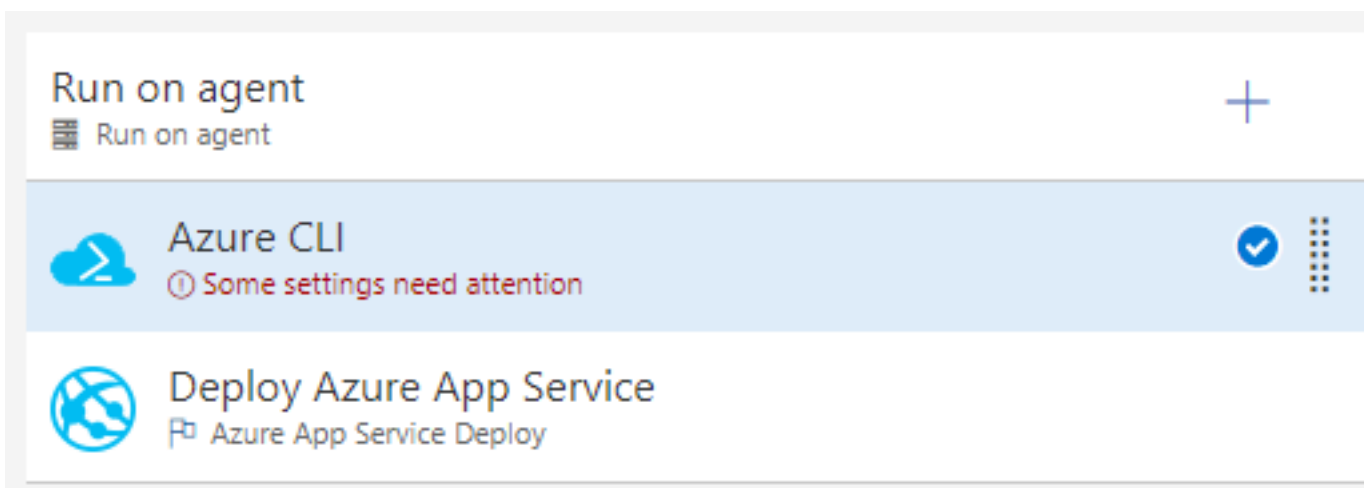
Create a Azure Web App

Now we have a pipeline that would deploy our application towards Azure. But this would fail while we have not created any infrastructure yet. To be able to succeed we are going to add tasks to our release to provision the Azure Web App to deploy to. We are using our already created batch files. If not created the files are attached to the workitem.

1. Add the Azure CLI Task to the pipeline



1. Drag it above the "Deploy Azure App Service" task




1. Configure the task, select the correct subscription and select the correct deployment script

Azure CLI ⓘ ✕ Remove

Version 1.* ▾

Display name *
Azure CLI

Azure subscription * ⓘ | [Manage](#)  [+ New](#)

GDBC ▾

Script Location * ⓘ
Script path ▾

Script Path * ⓘ

...

This setting is required.

Argument

Advanced

Working directory

Fail on error

Control

Output

Select File Or Folder

- Linked Artifacts
 - _Application (Build)
 - _DevOpsResources (Git)**
 - .gitignore
 - CreateResourceGroup.bat**
 - DeleteResourceGroup.bat
 - README.md

The artifacts published by each version will be available for deployment in Release Management. The last successful version of _DevOpsResources (Git) published the following artifacts: *.gitignore, CreateResourceGroup.bat, DeleteResourceGroup.bat and 1 more.*

Location

OK Cancel

1. The script contains the following content

```
""bash REM create resource group call az group create --name "xjg-rg-sampleapp" --location "West Europe"
```

```
REM create appservice plan call az appservice plan create -g "xjg-rg-sampleapp" -n "xjg-app-sampleapp" --sku F1
```

```
REM create azurewebapp call az webapp create -g "xjg-rg-sampleapp" -p "xjg-app-sampleapp" -n "xjg-webapp-sampleapp" ``
```

Note: You can specify your own preferred naming here. Note that your WAP_NAME, needs to be globally unique. Region needs to be a valid Azure Datacenter name, use the following link to find the correct values.

<https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-supported-services>

1. Now we automated created of a resource group and Azure Web App.

Remove the Azure Web App

Now that we have the environment being created we also want to remove it after creation to minimize costs. 1. Add another Azure CLI Task to the pipeline 2. Configure the task to run the "DeleteResourceGroup.bat" file. This contains the following script.

```
bash REM delete resource group call az group delete --name "xjg-rg-sampleapp" --yes
```

Note: When the Replace Tokens script runs the variables in this script are also replace while we target ALL *.bat files there. If you specify a specific script there you need to do it twice.

Run your Release

You have now prepared your release to create the needed Azure Resources, deploy your application and remove it and its resources after creation. 1. Queue a new release to see if your release runs. 2. Validate if your CI/CD pipeline is correctly configured by validating if a build runs after a code change, and after successful build, a release is being executed.