

Activate Azure with DevOps

Module 06: End-to-End DevOps - Lab 03 - Multistage YAML Pipelines

Student Lab Manual

Conditions and Terms of Use

Microsoft Confidential - For Internal Use Only

This training package is proprietary and confidential and is intended only for uses described in the training materials. Content and software is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or software included in such packages is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URLs and other Internet Web site references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Copyright and Trademarks

© 2020 Microsoft Corporation. All rights reserved.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see **Use of Microsoft Copyrighted Content** at <https://www.microsoft.com/en-us/legal/intellectualproperty/permissions/default>

Microsoft®, Internet Explorer®, and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

Parts of this lab has been taken from <https://azuredevopslabs.com/labs/azuredevops/yaml/>. View additional publicly available labs at <https://azuredevopslabs.com/>.

Contents

Introduction

Prerequisites

Exercise 1: Add stages to YAML Pipelines

Exercise 2: Download pipeline artifacts for the Deploy stage

Exercise 3: Create Azure resources using YAML pipeline

Exercise 4: Add task in the YAML pipeline to deploy the Web App

Exercise 5: Review the deployed site

Lab 6.3.1: End-to-End DevOps: Deploy using Multi-Stage YAML

Introduction

In this lab, you will create a pipeline using YAML to build the code, run the unit tests, create the infrastructure in Azure and deploy the app.

You'll learn:

- The basic features of multistage YAML pipelines
- Deploying infrastructure as code
- Understand the value of pipelines as code

Prerequisites

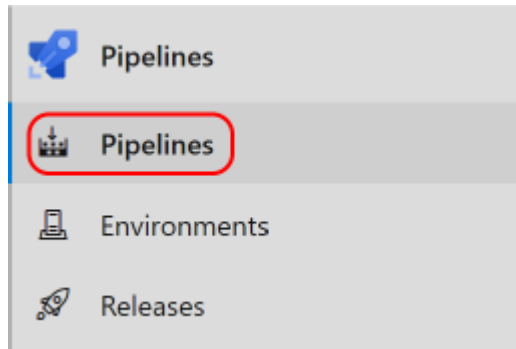
- Microsoft Azure subscription <https://azure.microsoft.com/>
- Lab 6.2 YAML Pipelines
- **Note:** You may want to disable other CI pipelines before starting the lab to prevent triggering multiple builds at once and causing agent delays in the queue
- If you've set up any branch policies on the master branch from the earlier labs, you may also need to temporarily disable those

Estimated Time To Complete This Lab

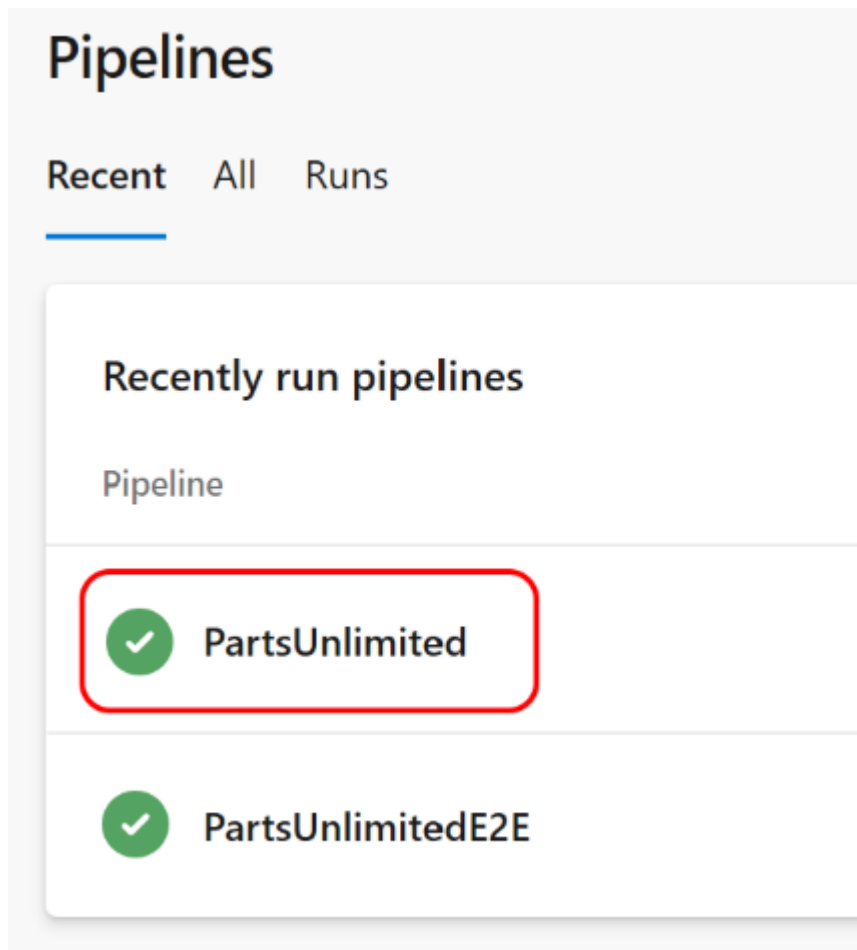
60 minutes

Exercise 1: Add stages to YAML Pipelines

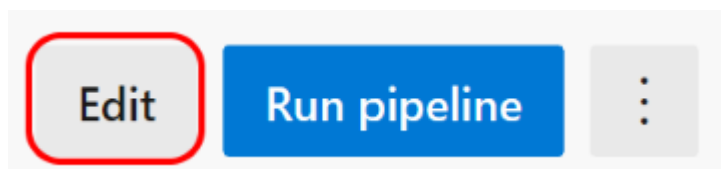
1. Navigate to the project used in the previous (YAML Pipeline) lab in Azure DevOps.
2. Navigate to **Pipelines**, then **Pipelines**.



3. Select and click on the YAML pipeline that we created in the build lab.



4. From the menu on the right, select Edit.



5. Now, first we will disable the Continuous Integration trigger so that saving the YAML file every time in this lab will not automatically trigger running of the pipeline. Change the **trigger** value to **none** like below:

```
trigger:
- none
```

master

PartsUnlimited / azure-pipelines-1.yml *

```
1 # ASP.NET
2 # Build and test ASP.NET projects.
3 # Add steps that publish symbols, save build artifacts, deploy, and more:
4 # https://docs.microsoft.com/azure/devops/pipelines/apps/aspnet/build-aspnet-4
5
6 trigger:
7   - none
8
9 pool:
10  - name: Hosted-VS2017
11    demands:
12      - msbuild
13      - visualstudio
14      - vstest
--
```

6. We will add the stage configuration lines below after the **trigger** section to define a **Build** stage in the YAML pipeline. You can define any stages you need to better organize and track pipeline progress. It's important that everything lines up as shown in the screenshot below. Remove extra spaces if necessary.

```
stages:
- stage: Build
  jobs:
  - job: Build
```

```
8
9 stages:
10  - stage: Build
11    jobs:
12      - job: Build
13
14  pool:
```

7. Highlight the remainder of the YAML file and indent it four spaces (press Tab button twice). This will simply take the existing build definition and relocate it as a child of the **jobs** node.

```

9   stages:
10  - stage: Build
11    jobs:
12    - job: Build
13
14      pool:
15        name: Hosted VS2017
16        demands:
17          - msbuild
18          - visualstudio
19          - vstest
20
21      variables:
22      - solution: '**/*.sln'

```

8. At the bottom of the file, add the configuration below to define a second stage.

```

- stage: Deploy
  jobs:
  - job: Deploy
    pool:
      name: Hosted VS2017
    steps:

```

```

Settings
51  | | | - task: PublishPipelineArtifact@1
52  | | |   inputs:
53  | | |     targetPath: '$(Build.ArtifactStagingDirectory)'
54  | | |     artifact: 'drop'
55  | | |     publishLocation: 'pipeline'
56
57  - stage: Deploy
58    jobs:
59    - job: Deploy
60      pool:
61        name: Hosted VS2017
62      steps:

```

9. In this exercise we have modified the existing YAML pipeline created in the previous lab to divide it into two stages: **Build** and **Deploy**. In the next exercise, we will start adding tasks and steps under the newly defined Deploy stage.

Exercise 2: Download pipeline artifacts for the Deploy stage

1. It's important to note that the two stages created in the previous exercise will run independently. As a result, the output from the first stage (Build) will not be available to the second stage (Deploy) without special consideration.

For this, we should already have a task that publishes the pipeline artifacts at the end of the build stage from the previous lab. We need to add another task to download these artifacts at the beginning of the deploy stage. Check your build stage to ensure you have a **Publish Pipeline Artifact** task at the end of the build stage.

```

51 |         - task: PublishPipelineArtifact@1
52 |           inputs:
53 |             targetPath: '$(Build.ArtifactStagingDirectory)'
54 |             artifact: 'drop'
55 |             publishLocation: 'pipeline'
56 |
57 |     - stage: Deploy
58 |       jobs:
59 |       - job: Deploy
60 |
61 |         pool:
62 |           name: Hosted VS2017

```

2. Place the cursor on the first line under the **steps** node of the deployment stage.

```

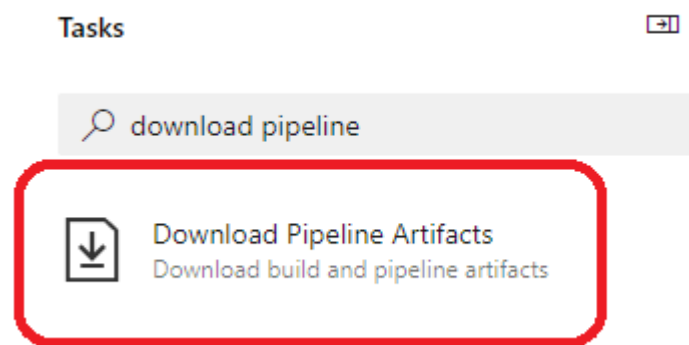
- stage: Deploy
  jobs:
  - job: Deploy

    pool:
      name: Hosted VS2017

    steps:

```

3. Search the tasks for **download pipeline** and select the **Download Pipeline Artifacts** task.



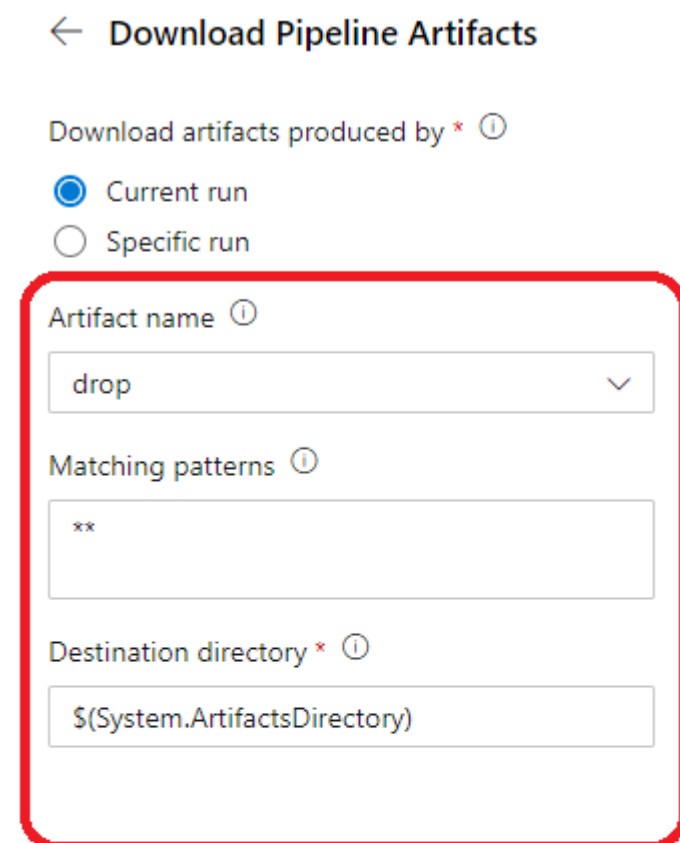
4. Set following values to the fields and then and click **Add**:

Download artifacts produced by: Current run

Artifact name: drop

Matching patterns: **

Destination directory: \$(System.ArtifactsDirectory)



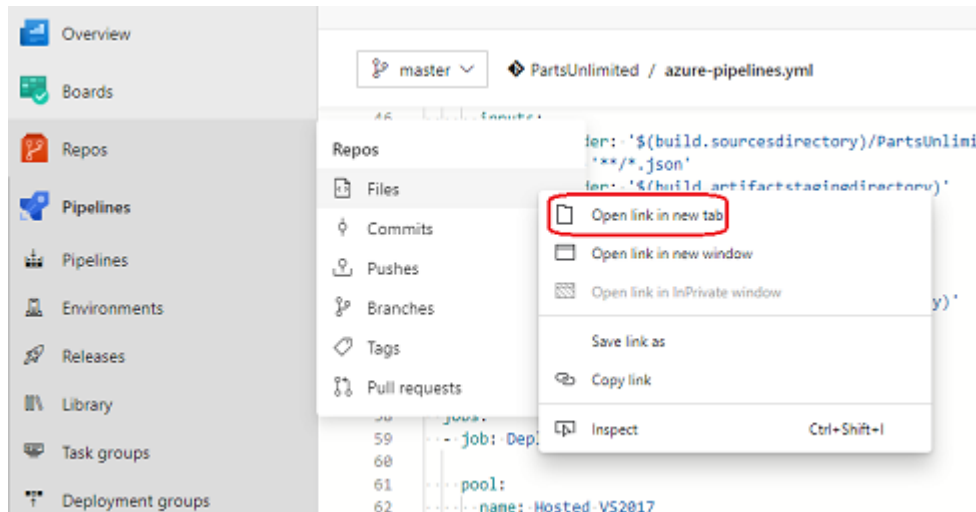
5. Indent the download task four spaces (two tabs) if it isn't already. You may also want to add an empty line before and after to make it easier to read.

```
57   - stage: Deploy
58     jobs:
59     - job: Deploy
60       pool:
61       name: Hosted VS2017
62       steps:
63         Settings
64         - task: DownloadPipelineArtifact@2
65           inputs:
66             buildType: 'current'
67             artifactName: 'drop'
68             targetPath: '$(System.ArtifactsDirectory)'
```

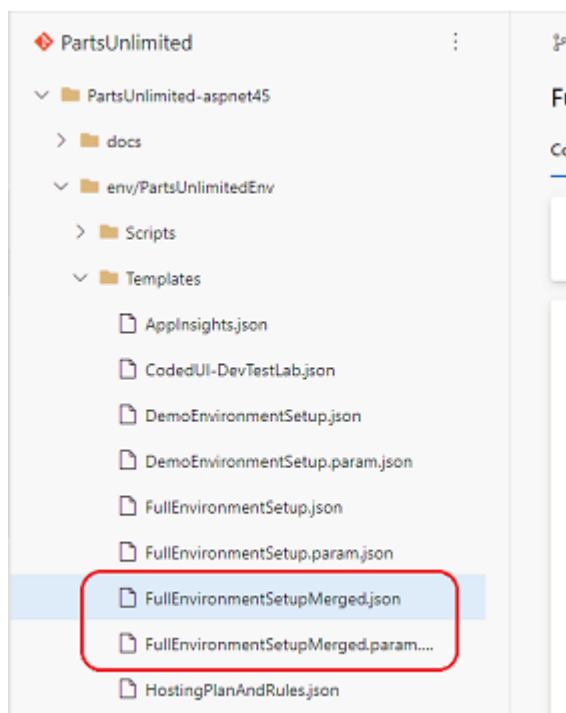
6. In the next exercise we will now create the Azure resources using the JSON files downloaded in this task.

Exercise 3: Create Azure resources using YAML pipeline

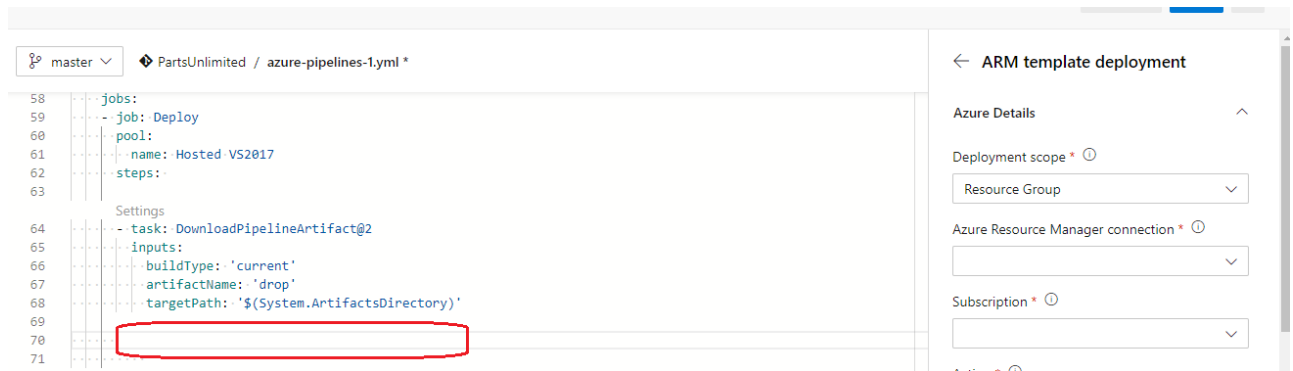
1. In this exercise, we will create required Azure resources (Azure Web App, Azure SQL Database, etc.) by using the ARM Template available in the repository and published and downloaded in the previous exercises.
2. Move the mouse pointer on **Repos** and right-click on **Files**. Click on **Open link in new tab** to open repository in a new browser tab.



3. Navigate to **PartsUnlimited-aspnet45/env/PartsUnlimitedEnv/Templates** and locate **FullEnvironmentSetupMerged.json** and **FullEnvironmentSetupMerged.param.json**. These are the two files we will use to build our Azure resources. Review these files.

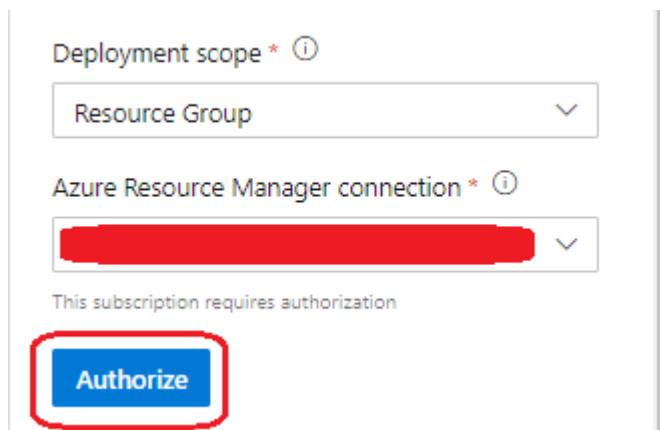


4. Switch back to the browser tab for editing Pipelines. Place the cursor on the next empty line in the YAML file. In the **Tasks**, search for **ARM template deployment** task and click on it.



5. This is the task that will take the ARM templates mentioned above and deploy Azure resources using those templates. Enter following information in this task:

- **Deployment scope:** Resource Group
- **Azure Resource Manager Connection:** From the drop down, select the right subscription
Note: If Authorization is requested, click **Authorize** and follow the path to complete authorization.



6. Once authorized successfully continue with following information:

- **Subscription:** Select the correct subscription
- **Action:** Create or update resource group
- **Resource group:** PartsUnlimitedRG (This resource group will be created when the task runs)
- **Location:** East US
- **Template location:** Linked artifact
- **Template:** \$(System.ArtifactsDirectory)/**/FullEnvironmentSetupMerged.json
- **Template parameters:** \$(System.ArtifactsDirectory)/**/FullEnvironmentSetupMerged.param.json
- **Override template parameters:** -WebsiteName \$(WebsiteName) -PUL_ServerName \$(ServerName) -PUL_DBPassword \$(AdminPassword) -PUL_DBPasswordForTest \$(AdminTestPassword) -PUL_HostingPlanName \$(HostingPlan)
- **Deployment mode:** Incremental
- Click **Add**

← ARM template deployment

Azure Details ^

Deployment scope * ⓘ

Resource Group

Azure Resource Manager connection * ⓘ

Visual Studio Ultimate with MSDN([REDACTED])

Subscription * ⓘ

Visual Studio Ultimate with MSDN ([REDACTED])

Action * ⓘ

Create or update resource group

Resource group * ⓘ

PartsUnlimitedRG

Location * ⓘ

East US

Template ^

Template location *

Linked artifact v

Template * ⓘ

\$(System.ArtifactsDirectory)/**/*.FullEnvironme

Template parameters ⓘ

\$(System.ArtifactsDirectory)/**/*.FullEnvironme

Override template parameters ⓘ

-WebsiteName \$(WebsiteName) -
 PUL_ServerName \$(ServerName) -
 PUL_DBPassword \$(AdminPassword) -
 PUL_DBPasswordForTest
 \$(AdminTestPassword) -
 PUL_HostingPlanName \$(HostingPlan)

Deployment mode * ⓘ

Incremental v

About this task

Add

7. Your YAML file should now looks like this:

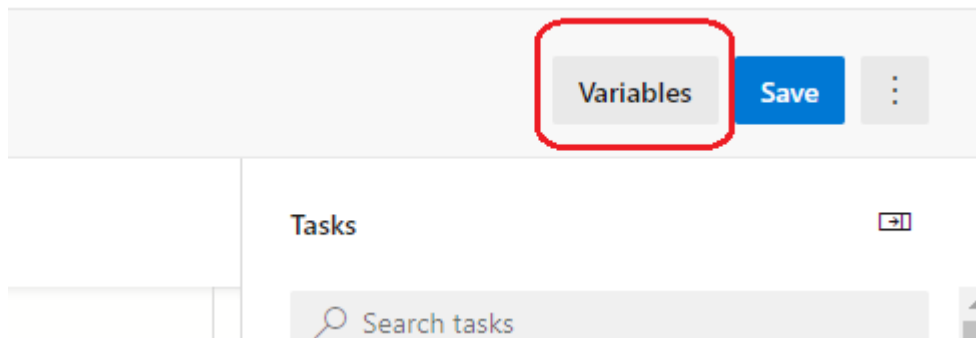
```

59 |     - job: Deploy
60 |       pool:
61 |         name: Hosted-VS2017
62 |       steps:
63 |
64 |       Settings
65 |       - task: DownloadPipelineArtifact@2
66 |         inputs:
67 |           buildType: 'current'
68 |           artifactName: 'drop'
69 |           targetPath: '$(System.ArtifactsDirectory)'
70 |
71 |       Settings
72 |       - task: AzureResourceManagerTemplateDeployment@3
73 |         inputs:
74 |           deploymentScope: 'Resource Group'
75 |           azureResourceManagerConnection: 'Visual Studio Ultimate with MSDN (XXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX)'
76 |           subscriptionId: 'XXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX'
77 |           action: 'Create Or Update Resource Group'
78 |           resourceGroupName: 'PartsUnlimitedRG'
79 |           location: 'East US'
80 |           templateLocation: 'Linked artifact'
81 |           csmFile: '$(System.ArtifactsDirectory)/**/*.FullEnvironmentSetupMerged.json'
82 |           csmParametersFile: '$(System.ArtifactsDirectory)/**/*.FullEnvironmentSetupMerged.param.json'
83 |           overrideParameters: '-WebsiteName $(WebsiteName) -PUL_ServerName $(ServerName) -PUL_DBPassword $(AdminPa
84 |           deploymentMode: 'Incremental'

```

Note: In case your sections are 'overindented' then you can select that entire section and press **Shift + Tab** twice to unindent the selected section.

8. We need to define the variables that we used in the previous step for overriding the template parameters. Click on **Variables** at the top-right of the screen.



9. Click on **New Variable** to add New Variable. Add **Name** as WebsiteName and **Value** as the name you intend to use for your website.

Note: Remember, the website name needs to be unique across all of Azure. You can achieve this by adding your initials at the end of the name.

← Update variable

Name

Value

☐ Keep this value secret

☐ Let users override this value when running this pipeline

To reference a variable in YAML, prefix it with a dollar sign and enclose it in parentheses. For example: `$(WebsiteName)`

To use a variable in a script, use environment variable syntax. Replace `.` and space with `_`, capitalize the letters, and then use your platform's syntax for referencing an environment variable. Examples:

Batch script: `%WEBSITENAME%`

PowerShell script: `$env:WEBSITENAME`

Bash script: `$WEBSITENAME`

10. Now we will define all the remaining variables mentioned in **Override template parameters** section above:

- **ServerName:** partsUnlimited-yaml-db-srv-[your initials] (Server name has to be unique across all of Azure. Add your initials in the server name to make it unique)
- **AdminPassword:** Pa\$\$w0rd (You can use any password you want and also check the box for **Keep this value secret**)
- **AdminTestPassword:** Pa\$\$w0rd (You can use any password you want and also check the box for **Keep this value secret**)
- **HostingPlan:** PUL-YAML
- Click **Save**

Note: If you decide to use some other password of your choice please make sure it meets following requirements:

- * Minimum of 8 characters
- * Requires 3 out of 4 of the following:
 - ** Lowercase characters
 - ** Uppercase characters
 - ** Numbers (0-9)
 - ** Symbols
- * Cannot contain the login name

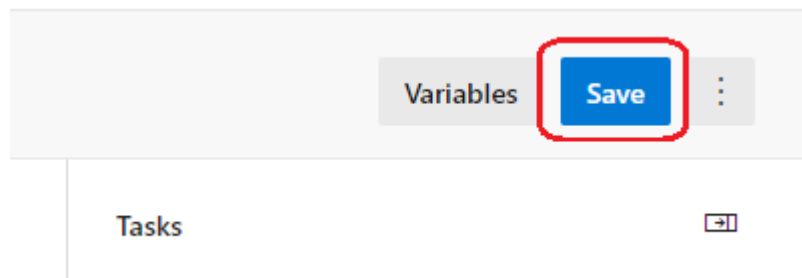
Variables



	AdminPassword	*****
	AdminTestPassword	*****
	HostingPlan	= PUL-YAML
	ServerName	= partsUnlimited-yaml-db-srv-kp
	WebsiteName	= PUL-YAML-Multistage-kp

[Learn about variables](#)[Cancel](#)[Save](#)

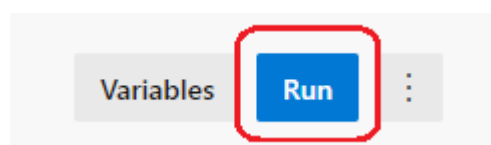
11. At this point it'd be best to check if the above task creates the environment in Azure as we expect. We will save and run the pipeline before continuing to the next exercise. Click on **Save** at the top-right.



12. Add a commit message and click **Save**.

A screenshot of the 'Save' dialog box in Azure Pipelines. The dialog has a title bar with a close button (X). Below the title, it says 'Saving will commit azure-pipelines-3.yml to the repository.' There is a text input field for the 'Commit message' with the text 'Update azure-pipelines-3.yml for Azure Pipelines' entered. Below this is a larger text area for the 'Optional extended description' with the placeholder text 'Add an optional description...'. At the bottom, there are two radio buttons: 'Commit directly to the master branch' (which is selected) and 'Create a new branch for this commit'. At the bottom right, there are two buttons: 'Cancel' and 'Save', with the 'Save' button highlighted by a red rectangular box.


13. Click on **Run** at the top-right corner and click **Run** again without making any changes.



Run pipeline

Select parameters below and manually run the pipeline

Branch/tag

 master

Select the branch, commit, or tag

Advanced options

Variables

This pipeline has no defined variables

Stages to run

Run as configured

Resources

Use latest version of all resources

☐ Enable system diagnostics

Cancel

Run

14. Click on **Pipelines** and view the recent run triggered. Click on the run to view more details.

Note: As the Build stage finishes and Deploy stage starts, you maybe asked to **View** and provide permissions to Azure resources. Click on **Permit**.



Boards

Repos

Pipelines

Pipelines


Recently run pipelines

Pipeline	Last run
 PartsUnlimited	#20200424.1 • Update azure-pipelines-3.yml for Azure Pipelines Manually triggered for  master



#20200424.1 Update azure-pipelines-3.yml for Azure Pipelines

on PartsUnlimited

Summary Tests Environments

Manually run by 

View 68 changes


Repository and version	Time started and elapsed	Related	Tests and coverage
 PartsUnlimited master 28ce5ac	Today at 4:14 PM 2m 18s	0 work items 1 published	 Get started


Warnings 1

PartsUnlimited-aspnet45\src\PartsUnlimitedWebsite\Controllers\ManageController.cs(135,41): Warning CS1998: This async method lacks 'await' operators and will run synchronously. Co...

Build • Build • VSBuild

Stages

 Build
1 job completed 1m 36s
87.5% tests passed
1 artifact

 DeployToDev
0/1 completed 31s
Cancel

Note: You can also see the two stages (Build and Deploy) of the pipeline here.

15. Once the pipeline run is finished, you can see the summary like this:

← **Jobs in run #20200424.4**
PartsUnlimited (1)

Build

- > ✔ Build 1m 36s

Deploy

- ✓ ✔ Deploy 3m 12s
 - ✓ Initialize job 4s
 - ✓ Checkout PartsUnlimite... 8s
 - ✓ DownloadPipelineArtif... 37s
 - ✓ AzureResourceMa... 2m 20s
 - ✓ Post-job: Checkout Pa... <1s
 - ✓ Finalize Job <1s

Finalize build

- ✓ Report build status <1s

✔ **AzureResourceManagerTemplateDeployment** View raw log

```

1 Starting: AzureResourceManagerTemplateDeployment
2 =====
3 Task      : ARM template deployment
4 Description : Deploy an Azure Resource Manager (ARM) template to all the deployment scopes
5 Version    : 3.2.0
6 Author     : Microsoft Corporation
7 Help       : https://docs.microsoft.com/azure/devops/pipelines/tasks/deploy/azure-resource-group-deployment
8 =====
9 ARM Service Connection deployment scope - Subscription
10 Checking if the following resource group exists: PartsUnlimitedRG.
11 Resource group exists: true.
12 Creating deployment parameters.
13 The detected encoding for file 'D:\a\1\PartsUnlimitedEnv\Templates\FullEnvironmentSetupMerged.json' is 'utf-8'
14 The detected encoding for file 'D:\a\1\PartsUnlimitedEnv\Templates\FullEnvironmentSetupMerged.param.json' is 'utf-8'
15 Starting template validation.
16 Deployment name is FullEnvironmentSetupMerged-20200424-040547-398e
17 Template deployment validation was completed successfully.
18 Starting Deployment.
19 Deployment name is FullEnvironmentSetupMerged-20200424-040547-398e
20 Successfully deployed the template.
21 Finishing: AzureResourceManagerTemplateDeployment
          
```

16. Go to **Azure Portal** and navigate to the PartsUnlimitedRG that got created. It should list all the resources the deployment created:

PartsUnlimitedRG
Resource group

Search (Ctrl+/)

- Overview
- Activity log
- Access control (IAM)
- Tags
- Events

Settings

- Quickstart
- Deployments
- Policies
- Properties
- Locks
- Export template

Cost Management

- Cost analysis
- Cost alerts (preview)
- Budgets
- Advisor recommendations

Subscription (change): Visual Studio Ultimate with MSDN Deployments: 1 Failed, 4 Succeeded

Subscription ID: [REDACTED]

Tags (change): CostCenter: ContosoIT

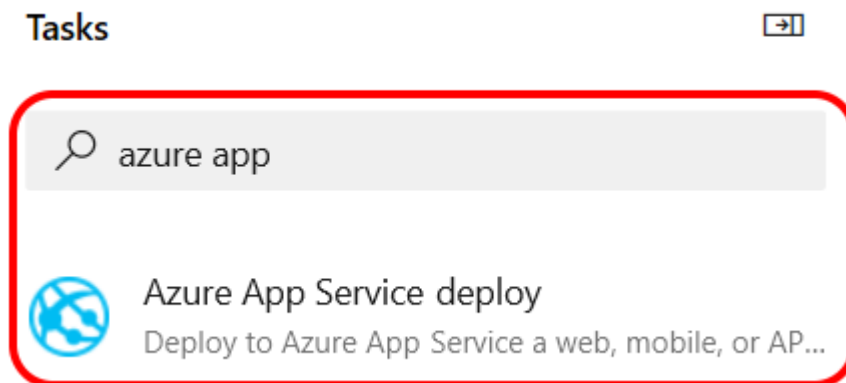
Filter by name... Type == all Location == all Add filter

Showing 1 to 13 of 13 records. Show hidden types No

Name ↑↓	Type ↑↓	Location ↑↓
partsunlimited-yaml-db-srv-kp	SQL server	East US
partsunlimited-yaml-db-srv-kpdev	SQL server	East US
partsunlimited-yaml-db-srv-kpstage	SQL server	East US
PartsUnlimitedDB (partsunlimited-yaml-db-srv-kp/PartsUnlimitedDB)	SQL database	East US
PartsUnlimitedDB (partsunlimited-yaml-db-srv-kpdev/PartsUnlimitedDB)	SQL database	East US
PartsUnlimitedDB (partsunlimited-yaml-db-srv-kpstage/PartsUnlimitedDB)	SQL database	East US
PUL-YAML-Multistage-kp	App Service	East US
PUL-YAML-Multistage-kp-DevInsights	Application Insights	East US
PUL-YAML-Multistage-kp-Insights	Application Insights	East US

Exercise 4: Add task in the YAML pipeline to deploy the Web App

1. Now that the environment is created in Azure, we will continue with adding the last task in the YAML pipeline to deploy the web app. Navigate back to **Azure DevOps** and to the project. Click on **Pipelines**, select the pipeline you were working on in the previous exercise. Click on **Edit Pipeline**.
2. Set the cursor on a new line at the end of the YAML definition. This will be the location where new tasks are added.
3. Search the tasks for "**Azure App**" and select the **Azure App Service Deploy** task.



4. Add following details in the task:
 - **Azure subscription:** Select the subscription used in the previous exercise
 - **App Service name:** \$(WebsiteName)
 - **Package or folder:** \$(System.ArtifactsDirectory)/**/*.zip
 - Click **Add**

← Azure App Service deploy

Connection type * ⓘ
 Azure Resource Manager

Azure subscription * ⓘ
 Visual Studio Ultimate with MSDN

App Service type * ⓘ
 Web App on Windows

App Service name * ⓘ
 \$(WebsiteName)

☐ Deploy to Slot or App Service Environment ⓘ

Virtual application ⓘ

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

File transforms & variable substitution Op... ✓

Additional Deployment Options ✓

Post Deployment Action ✓

About this task

Add

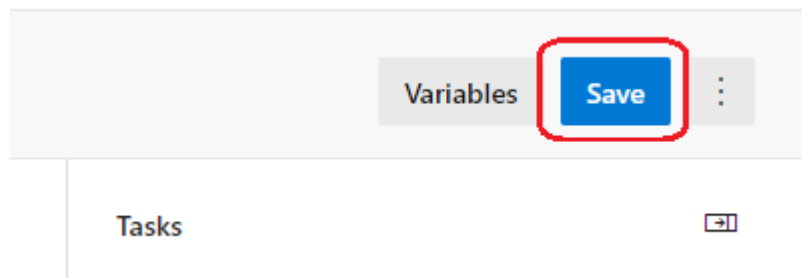
5. The YAML that defines the task will be added to the cursor location in the file. Ensure it is indented and thus a child of the **steps** task. If not, indent it four spaces (two tabs).

```

85 | |
86 | | Settings
87 | | - task: AzureRmWebAppDeployment@4
88 | |   inputs:
89 | |     - connectionType: 'AzureRM'
90 | |     - azureSubscription: 'Visual Studio Ultimate with MSDN(')
91 | |     - appType: 'webApp'
92 | |     - webAppName: '$(WebsiteName)'
93 | |     - packageForLinux: '$(System.ArtifactsDirectory)/**/*.zip'

```

6. Click **Save** to commit the changes.



7. Confirm the **Save** and **Run** the pipeline again.

Save

Saving will commit azure-pipelines.yml to the repository.

Commit message

Update azure-pipelines.yml for Azure Pipelines

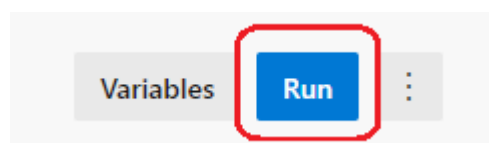
Optional extended description

Add an optional description...

- ☒ Commit directly to the master branch
☐ Create a new branch for this commit

Cancel

Save




Run pipeline

×

Select parameters below and manually run the pipeline

Branch/tag

 master

▼

Select the branch, commit, or tag

Advanced options

Variables

This pipeline has no defined variables

>

Stages to run

Run as configured

>

Resources

Use latest version of all resources

>

☐ Enable system diagnostics

Cancel

Run



8. Return to the **Pipelines** view and from the **Runs** tab, click the new run to open it.

Pipelines

New


Recent All Runs

All pipeline runs

Description	Stages	
Update azure-pipelines.yml for Azure Pipelines #20200326.9 on PartsUnlimited master 6d0e0e5		 Just now 14s


9. When the Build stage completes, click the **Deploy** stage to follow each task.


Stages Jobs

 Build

1 job completed


2m 56s

 1 artifact

 Deploy

0/1 completed

15s

 Deploy

15s

Cancel

10. Expand the **AzureRmWebAppDeployment** task to review the steps performed during the Azure deployment. Once the task completes, your app will be live on Azure.

←

Jobs in run #20200423.19

PartsUnlimited (1)

Build

>

✓ Build

2m 16s

Deploy

▼

✓ Deploy

2m 24s

✓ Initialize job

4s

✓ Checkout PartsUnlimited...

7s

✓ DownloadPipelineArtifact

2s

✓ AzureResourceMa...

1m 47s

✓ AzureRmWebAppDepl...

22s

✓ Post-job: Checkout Pa...

<1s

✓ Finalize Job

<1s

Finalize build

✓ AzureRmWebAppDeployment

```

1 Starting: AzureRmWebAppDeployment
2 =====
3 Task      : Azure App Service deploy
4 Description : Deploy to Azure App Service a web, mobile, or API app using Docker, Java, .NET, .NET Core, Node.
5 Version    : 4.163.5
6 Author     : Microsoft Corporation
7 Help       : https://aka.ms/azureappservicetroubleshooting
8 =====
9 Got service connection details for Azure App Service: 'PUL-YAML-Multistage-kp'
10 Trying to update App Service Application settings. Data: null
11 App Service Application settings are already present.
12 "C:\Program Files\IIS\Microsoft Web Deploy V3\msdeploy.exe" -verb:sync -source:package='D:\a\1\1\PartsUnlimited
13 Info: Using ID '9c0e762b-d5c2-4c85-98d2-908d1bf4f0b1' for connections to the remote server.
14 Info: Adding ACLs for path (PUL-YAML-Multistage-kp)
15 Info: Adding ACLs for path (PUL-YAML-Multistage-kp)
16 Info: Using ID '13f6839c-0a0e-4e54-924a-17d61ab9817f' for connections to the remote server.
17 Info: Updating file (PUL-YAML-Multistage-kp\ApplicationInsights.config).
18 Info: Updating file (PUL-YAML-Multistage-kp\Areas\Admin\Views\Customer\Find.cshtml).
19 Info: Updating file (PUL-YAML-Multistage-kp\Areas\Admin\Views\Customer\Index.cshtml).
20 Info: Updating file (PUL-YAML-Multistage-kp\Areas\Admin\Views\Orders\Details.cshtml).
21 Info: Updating file (PUL-YAML-Multistage-kp\Areas\Admin\Views\Orders\Index.cshtml).
22 Info: Updating file (PUL-YAML-Multistage-kp\Areas\Admin\Views\Raincheck\Index.cshtml).
23 Info: Updating file (PUL-YAML-Multistage-kp\Areas\Admin\Views\StoreManager\Create.cshtml).
24 Info: Updating file (PUL-YAML-Multistage-kp\Areas\Admin\Views\StoreManager\Details.cshtml).
25 Info: Updating file (PUL-YAML-Multistage-kp\Areas\Admin\Views\StoreManager\Edit.cshtml).
26 Info: Updating file (PUL-YAML-Multistage-kp\Areas\Admin\Views\StoreManager\Index.cshtml).

```

Exercise 5: Review the deployed site

1. Return to the Azure portal browser tab.
2. Navigate to the **PartsUnlimitedRG** resource group.
3. Click on **PUL-YAML-Multistage-kp** App Service

The screenshot shows the Azure portal interface for the 'PartsUnlimitedRG' resource group. The left sidebar contains navigation options like Overview, Activity log, Access control (IAM), Tags, Events, Settings, Quickstart, Deployments, Policies, Properties, Locks, Export template, Cost Management, Cost analysis, Cost alerts (preview), Budgets, and Advisor recommendations. The main pane displays a list of resources. The 'PUL-YAML-Multistage-kp' App Service is highlighted with a red rectangle. Above the list, there are filters for Name, Type, and Location, and a table showing 13 records. The table columns are Name, Type, and Location. The highlighted resource is 'PUL-YAML-Multistage-kp' of type 'App Service' located in 'East US'.

4. Click **Browse** to open your site in a new tab.

The screenshot shows the 'Browse' button highlighted with a red circle. Below the button, the resource group is listed as 'partsunlimited' and the status is 'Running'.

5. The deployed site should load as below.

The screenshot shows the deployed website 'Parts Unlimited' in a browser. The URL is 'https://pul-yaml-multistage-kp.azurewebsites.net'. The website features a navigation bar with links: Brakes, Lighting, Wheels & Tires, Batteries, Oil, and More. Below the navigation bar, there is a banner for 'New Arrival : Bugeye Headlights (2 Pack)' and a large text overlay that reads 'ALL OIL AND FILTERS'.

Note: Back in the **PartsUnlimitedRG** resource group, you might have noticed other App Services (Slot) and database servers with "dev" and "stage" suffixes. These can be used to deploy the application in multiple phases/rings where deployment first goes to Dev and then to Stage and then finally to the Production environment. We will explore this in the next lab.