Prerequisites

* Azure Subscription
* Azure DevOps Account and Team Project
* Azure DevOps Service Connection

**## Example**

You can find the sample ARM template on GitHub

<https://github.com/Azure/azure-quickstart-templates/tree/master/101-webapp-basic-linux>

You can always modify a template to meet your needs, but for this example we’ll use the ARM template below.

azuredeploy.json

{

"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"webAppName": {

"type": "string",

"metadata": {

"description": "Base name of the resource such as web app name and app service plan "

},

"minLength": 2

},

"sku":{

"type": "string",

"metadata": {

"description": "The SKU of App Service Plan "

}

},

"linuxFxVersion" : {

"type": "string",

"defaultValue" : "php|7.0",

"metadata": {

"description": "The Runtime stack of current web app"

}

},

"location": {

"type": "string",

"metadata": {

"description": "Location for all resources."

}

}

},

"variables": {

"webAppPortalName": "[concat(parameters('webAppName'), '-webapp')]",

"appServicePlanName": "[concat('AppServicePlan-', parameters('webAppName'))]"

},

"resources": [

{

"type": "Microsoft.Web/serverfarms",

"apiVersion": "2017-08-01",

"kind": "linux",

"name": "[variables('appServicePlanName')]",

"location": "[resourceGroup().location]",

"dependsOn": [],

"sku": {

"name": "[parameters('sku')]"

},

"properties": {

"reserved": true

}

},

{

"type": "Microsoft.Web/sites",

"apiVersion": "2016-08-01",

"kind": "app",

"name": "[variables('webAppPortalName')]",

"location": "[resourceGroup().location]",

"properties": {

"serverFarmId": "[resourceId('Microsoft.Web/serverfarms', variables('appServicePlanName'))]",

"siteConfig": {

"linuxFxVersion": "[parameters('linuxFxVersion')]"

}

},

"dependsOn": [

"[resourceId('Microsoft.Web/serverfarms', variables('appServicePlanName'))]"

]

}

]

}

You can provide parameter values in the parameters.json file, or have these values be overwritten at runtime in the Azure DevOps pipeline, which we’ll see later in this tutorial.

azuredeploy.parameters.json

{

"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"webAppName": {

"value": "linuxwebapptest4"

},

"sku": {

"value": "S1"

},

"linuxFxVersion": {

"value": "TOMCAT|9.0-jre8"

},

"location": {

"value": "centralus"

}

}

}

**Setting up Azure DevOps Pipeline**

To set up the build and release pipeline by the appropriate tasks:

MSBuild:

Agent pool = Hosted VS2017

Copy Files to:

Display name = Copy Files to: $(build.artifactstagingdirectory)

Source Folder = <Using the ellipses to select the source folder>

Contents = \*\*\\*.json

Target Folder = $(build.artifactstagingdirectory)

Publish Artifact: drop

Path to publish = $(Build.ArtifactStagingDirectory)

Artifact name = drop

Artifact publish location = Azure Pipelines/TFS

Release:

Azure Resource Group Deployment:

Azure subscription = <Azure Service Connection>

Action = Create or update resource group

Resource group = <Azure Resource group>

Location = <Azure Resource group location>

Template location = Linked artifact

Template = <Using the ellipses navigate to the drop folder and select the .json file>

Template parameters = <Using the ellipses navigate to the drop folder and select the parameters.json file>

Override template parameters = <Here you can set parameter values and override the values in the parameters file.>

Deployment mode = <Incremental>

Yaml versions:

resources:

- repo: self

pool:

vmImage: Hosted VS2017

demands: msbuild

steps:

- task: MSBuild@1

displayName: 'Build solution <Solution name/project name>'

inputs:

solution: '<Solution name/project name>'

- task: CopyFiles@2

displayName: 'Copy Files to: $(build.artifactstagingdirectory)'

inputs:

SourceFolder: '<Solution name/project name>'

Contents: '\*\*\\*.json'

TargetFolder: '$(build.artifactstagingdirectory)'

CleanTargetFolder: true

OverWrite: true

- task: PublishBuildArtifacts@1

displayName: 'Publish Artifact: drop'

There are several options for selecting a repository source. For this example we’ll select Azure Repos Git and accept the defaults for the Team Project, Repository and Branch then select Continue.