# Provision Azure Resource Group and Deploy Application using the PowerShell DSC Resource Extension

Level: 200

In just a few steps, you can provision Virtual Machines (VMs) in Azure using [Resource Manager templates](https://azure.microsoft.com/en-in/documentation/articles/resource-group-template-deploy/) and start using them deploying and testing applications. This works from both the Visual Studio Online (VSO) service and your on-premises Team Foundation Server (TFS). The GitHub and the Azure service connections described here will be reused in other advanced tutorials.

## Connect to the GitHub Repository

The template for creating the Azure VMs is in the FabrikamFiber GitHub repository. Connect the repository to the VSO/TFS project to use the template directly from there.

Open your team project in your web browser.

* + On-premises http://{your\_server}:8080/tfs/DefaultCollection/{your\_team\_project}
  + Visual Studio Online https://{your\_account}.visualstudio.com/DefaultCollection/{your\_team\_project}

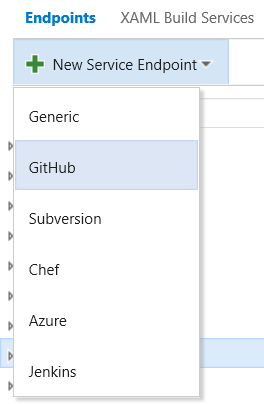
Click the gear icon on the top right to open the administrative control panel.



Click on the Service tab and then on Endpoints.



Click on the dropdown - New Service Endpoint and select GitHub.

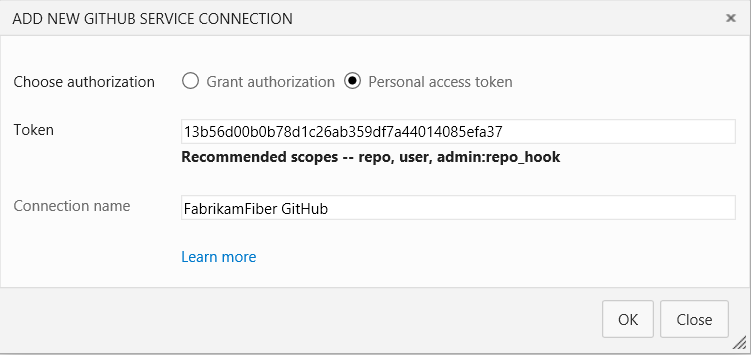


Fill in the parameters as shown below. Note that the Connection Name can be anything that you like but the token has to be as listed below. After entering the parameters click on OK. After this the source code repository at <https://github.com/fabrikamfiber> will be available in Build definitions.

**Choose Authorization:** Personal access token

**Token:** 13b56d00b0b78d1c26ab359df7a44014085efa37

**Connection Name:** FabrikamFiber GitHub



On clicking OK, you will see the FabrikamFiber GitHub connection in the endpoints panel on the left.

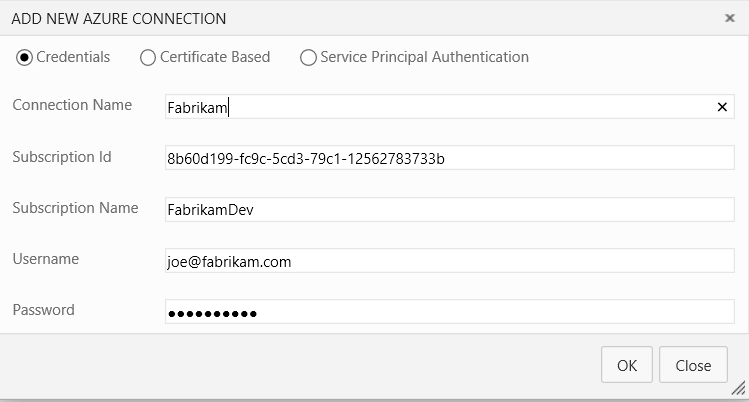
## Connect to the Azure Subscription

To connect to an Azure Subscription, select Azure from the New Service Endpoint as shown above. On the Add Azure Subscription dialog box:

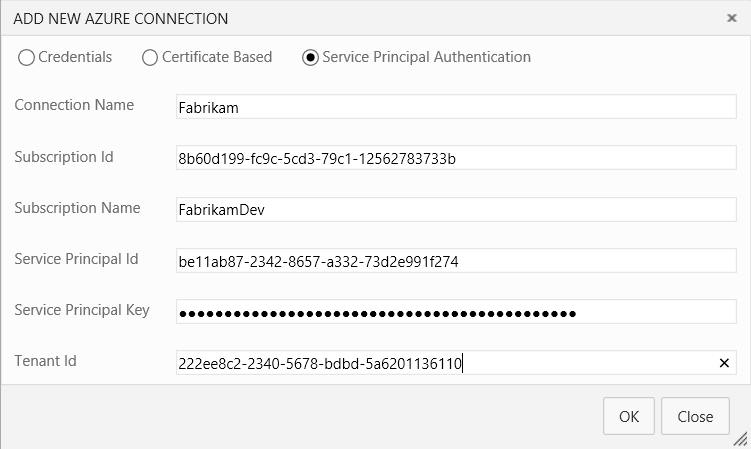
1. Select Credentials and enter a user friendly name for the connection name like Fabrikam.
2. Log into the [Azure portal](https://portal.azure.com/) and click on Subscriptions.
3. Copy the subscription ID and the subscription name and paste them into the Add Azure Subscription dialog box.
4. Enter a [work account’s](https://azure.microsoft.com/en-in/pricing/member-offers/msdn-benefits-details/work-accounts-faq/) username and password. Note that Live IDs like [joe@hotmail.com](mailto:joe@hotmail.com) or [joe@outlook.com](mailto:joe@outlook.com) are not supported.

If you have an Azure MSDN account, then you can easily create a work account by following the steps below:

1. Create an user in the Azure Active Directory from the [portal](https://msdn.microsoft.com/en-us/library/azure/hh967632.aspx) (this is the old Azure portal).
   * Add the Active Directory account to the co-administrators in the subscription. Go to the Settings and then click on administrators and add the account as a co-admin like, [testuser@joehotmail.onmicrosoft.com](mailto:testuser@joehotmail.onmicrosoft.com)
   * Login to the portal with this Active Directory account wiz. [testuser@joehotmail.onmicrosoft.com](mailto:testuser@joehotmail.onmicrosoft.com), and change the password. Initially a temporary password is created and that needs to be changed at the first login.
2. Add that user and password in the service connections in the VSO and deployments will work with that account.

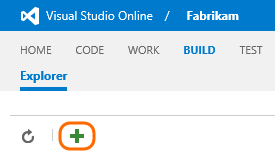


If there is no work account, then create a Service Principal as described [here](http://blogs.msdn.com/b/visualstudioalm/archive/2015/10/04/automating-azure-resource-group-deployment-using-a-service-principal-in-visual-studio-online-build-release-management.aspx) and enter the details as shown below:



## Create the build definition

Click on the tab titled Build and create a build definition.



On the Definition Templates dialog box, and click empty to start with.

## Add build steps

In the build definition, add some steps.

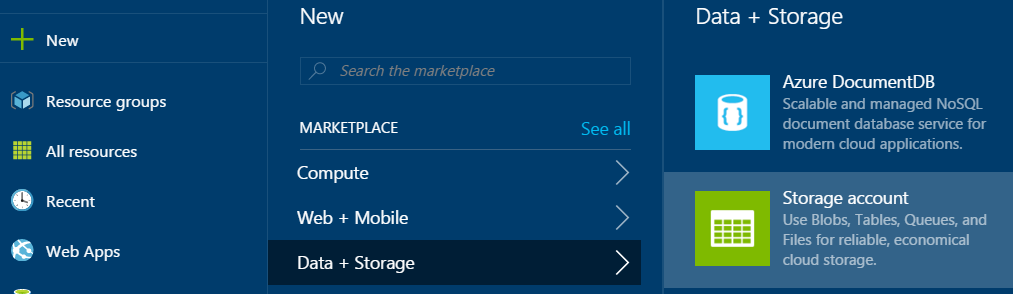
|  |  |
| --- | --- |
| C:\Users\ambrishm\AppData\Local\Microsoft\Windows\INetCache\Content.Word\RepoTab.png | **Setup the GitHub repository to use in the Build Definition**  Click on the Repository tab in the Build Definition and select the repository type as GitHub. Select the FabrikamFiber GitHub connection from the dropdown, and then select the repository as fabrikamfiber/marketing from the dropdown. Leave the default branch to master and the clean option to false. |
|  | **Setup the Variables to use in the Build Definition**  We need to define the administrator username and password for the virtual machines and as they are secrets, we will use the variables for them.  Click on the Variables tab in the Build Definition and then click on the Add Variable link twice. Enter the name and value pairs as shown below:   |  |  | | --- | --- | | **Name** | **Value** | | Configuration | release | | Platform | Any CPU | |
|  | **Build Solution using MSBuild**  Click on the Add build step and from the build tasks, select the MSBuild task. Fill in the parameters of the task as described below and let others be at their default values:  **Project**  Use the Source Control picker to select the solution file and the path will Marketing.sln  **Platform**  Enter the variable $(Platform)  **Configuration**  Enter the variable $(Configuration)  **MSBuild Arguments**  Enter the following arguments:  /p:DeployOnBuild=true /p:WebPublishMethod=Package /p:PackageAsSingleFile=true /p:SkipInvalidConfigurations=true /p:PackageLocation="$(build.sourcesDirectory)\pkg" |
|  | **Copy and Publish Build Artifacts**  Click on the Add build step and from the utility tasks, select the Copy and Publish Build Artifacts tasks. Fill in the parameters of the task as described below and let others be at their default values:  **Copy Root**  Enter the system variable – $(build.sourcesDirectory)  **Contents**  Enter the content to copy as listed below:  \*\*\pkg  \*\*\Deployment  **Artifacts Name**  Name the artifact – Fabrikam  **Artifacts Type**  Select Server from the dropdown |

## Build the definition

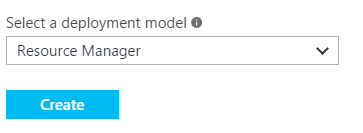
Save your definition as Fabrikam.Marketing.CI and queue the build on the hosted agent. Once the build is successful, follow the steps below to deploy the build using Release Management.

## Create an Azure storage account

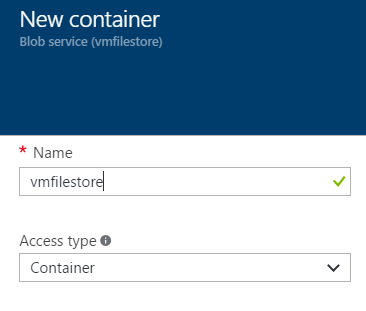
For the walkthrough, an Azure storage account is required. Log into the new [Azure portal](https://portal.azure.com/). Click on the +New from the left panel and then on Data + Storage. Select Storage Account from the list.



Select the deployment model as Resource Manager from the bottom of the Storage Account blade and click the Create button.

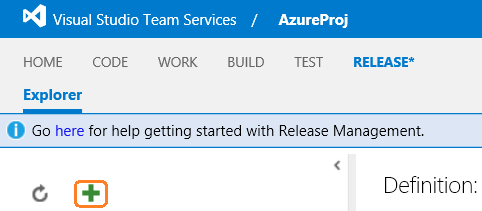


In the Create Storage Account blade fill in the name of the account like vmfilestore and the name of the resource group, and leave all others at their default values, and click on the Create button. Once the storage account has been created, open its blade, and click on the Blob tile. In the Blob service blade click on the Container tile, and in the Container blade, click on the +Container on the top to create a container. In the New Container blade, type the name of the container like psdscdemo, and for the Access type field select Container from the dropdown. Click on the Create button at the bottom of the blade to create the new container.



## Create the release definition

Click on the tab titled Release and create a release definition.



On the Deployment Templates dialog box, and click empty to start with. Click on the environment titled Default Environment and change it to Staging and save the release definition as Fabrikam.Marketing.Release.

## Add release steps

In the release definition, add some steps.

|  |  |
| --- | --- |
|  | **Setup the Artifacts Source in the Release Definition**  Click on the Artifacts tab, and then click on the button titled - Link an artifact source. In the pop-up window select the Build Definition created above and then click on the Link button. |
|  | **Setup the Variables to use in the Release Definition**  We need to define the VMs administrator username & password, and the Database username & password and as they are secrets, we will use the variables for them. Note that the usernames and passwords shown here have to be same as that for can be anything that you want and the ones shown below are for representational purposes only. After entering the name-value pairs, click on the lock icon next to them. This will store them securely and they will not be visible to anyone else.  In addition, we also need the URI to download the PowerShell script and the Web Deploy package. The URI that is shown below is constructed from the storage account name wiz. vmfilestore, and the container name wiz. psdesdemo. If you have used some other names for the storage account and the container, then replace those in the URI strings below.  Click on the Configuration tab in the Release Definition and then click on the Add Variable link five times. Enter the name and value pairs as shown below:   |  |  | | --- | --- | | **Name** | **Value** | | adminUsername | vmuser | | adminPassword | Password~111@ | | dbLogin | dbuser | | dbPassword | DBPassword~123@ | | modulesUrl | https://vmfilestore.blob.core.windows.net/psdscdemo/ConfigureWebServer.ps1.zip | | webDeployPackage | https://vmfilestore.blob.core.windows.net/psdscdemo/Marketing.zip | |
|  | **Setup the Triggers to use in the Release Definition**  Setup a Continuous Deployment trigger for the release definition. Click on the Triggers tab and then select the Continuous Deployment check-box. Select the Fabrikam.Marketing.CI from the dropdown in the – Set trigger on artifact source label. |
|  | **Copy files to Azure Blob**  Click on the Add tasks and from the Deployment tasks, select the Azure File Copy task. Fill in the parameters of the task as described below and let others be at their default values:  **Source**  Use the artifact picker to navigate to the PowerShell file (ConfigureWebServer.ps1.zip) to configure the Web Server. This will be only available if the source code was built and published to the VSO/TFS server. The value in the field will be - $(System.DefaultWorkingDirectory)\Fabrikam.Marketing.CI\Fabrikam\Marketing\Deployment\ConfigureWebServer.ps1.zip  **Azure Subscription**  Select the Azure subscription from the dropdown  **Destination Type**  Select Azure blob.  **Storage Account**  Enter the name of the storage account that was created earlier wiz. vmfilestore.  **Container Name**  Enter the name of the Container that was created earlier wiz. psdscdemo. |
| C:\Users\ambrishm\AppData\Local\Microsoft\Windows\INetCache\Content.Word\AzureFileCopy.png  C:\Users\ambrishm\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ArtifactPicker.png | **Copy files to Azure Blob**  Click on the Add tasks and from the Deployment tasks, select the Azure File Copy task. Fill in the parameters of the task as described below and let others be at their default values:  **Source**  Use the artifact picker to navigate to the Web deploy file (Marketing.zip) to install the website. This will be only available if the source code was built and published to the VSO/TFS server. The value in the field will be - $(System.DefaultWorkingDirectory)\Fabrikam.Marketing.CI\Fabrikam\pkg\Marketing.zip  **Azure Subscription**  Select the Azure subscription from the dropdown  **Destination Type**  Select Azure blob.  **Storage Account**  Enter the name of the storage account that was created earlier wiz. vmfilestore.  **Container Name**  Enter the name of the Container that was created earlier wiz. psdscdemo. |
|  | **Create an Azure Resource Group and Deploy the Application to the VMs**  Click on the Add tasks and from the deployment tasks, select the Azure Resource Group Deployment task. Fill in the parameters of the task as described below and let others be at their default values:  **Azure Subscription**  Select the Azure subscription from the dropdown  **Action**  Select – Create or Update Resource Group.  **Resource Group**  Enter the name of a Resource Group that does not exist in the subscription like, FabrikamMktg1. If the Resource Group already exists in the Subscription, then it will be updated with the resources specified in the template.  **Location**  Select West US from the dropdown.  **Template**  Use the Source Control picker to select the template file and the path will be -$(System.DefaultWorkingDirectory)\Fabrikam.Marketing.CI\Fabrikam\Marketing\Deployment\AzureVMDSCDeploy.json  **Template parameters**  Use the Source Control picker to select the template parameters file and the path will be – $(System.DefaultWorkingDirectory)\Fabrikam.Marketing.CI\Fabrikam\Marketing\Deployment\AzureVMDSCDeploy.params.json  **Override Template parameters**  The template will be used by many people and the unique resources like the storage accounts can conflict. These unique parameters will be overridden by inserting the line below. Change the names of the storage accounts and others by changing characters like xyges123 can be changed to gyxes321.  -newStorageAccountName xyges123 -vmName psdsc123 -adminUserName $(adminUsername) -adminPassword $(adminPassword) -dnsName psdsc123 -webdeploypkg $(webDeployPackage) -DatabaseServerName pscdsc123 -databaseServerAdminLogin $(dbLogin) -databaseServerAdminLoginPassword $(dbPassword) -databaseName psdsc123 -modulesUrl $(modulesUrl) |

## Build the definition

Save your release definition and then go to the build definition wiz. Fabrikam.Marketing.CI and queue the build on the hosted agent. Once the build is successful, it will automatically trigger the release definition. It will take 30 to 40 minutes to complete the deployment because it creates resource group, and then the VM in it, and creates an Azure SQL Server & a DB, and then deploys the application to the VM using the Azure’s resource extension for PowerShell-DSC. After the release completes, open the URL below in the browser to open the website. Note that the psdsc123 is the dnsName as specified in the override parameters above. If you have changed it then replace psdsc123 with the actual DNS name that you have supplied to the parameter.

http://psdsc123.westus.cloudapp.azure.com/Marketing\_deploy