# Provision Virtual Machines in Azure using Templates for Manual or UAT Testing

Level:100

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}

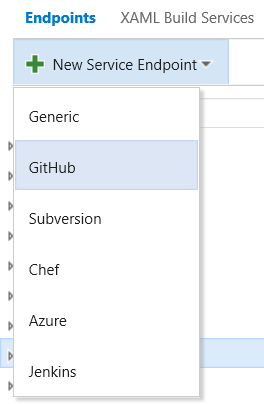
Click the gear icon on the top right to open the administrative control panel.



Click on the Service tab and then on Endpoints.



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

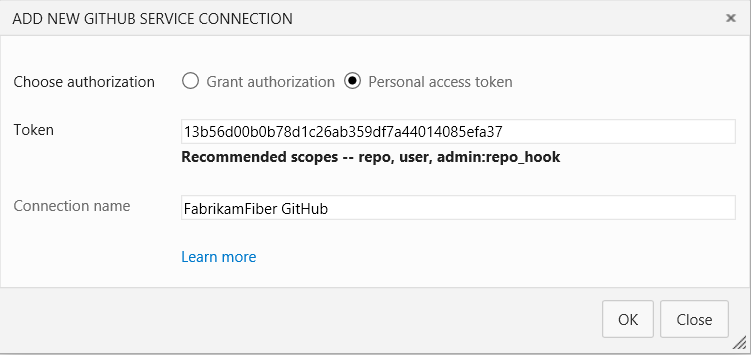


Fill in the parameters as shown below. Note that the Connection Name can be anything that you like but the token has to be as listed below. After entering the parameters click on OK. After this the source code repository at <https://github.com/fabrikamfiber> will be available in Build definitions.

**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.

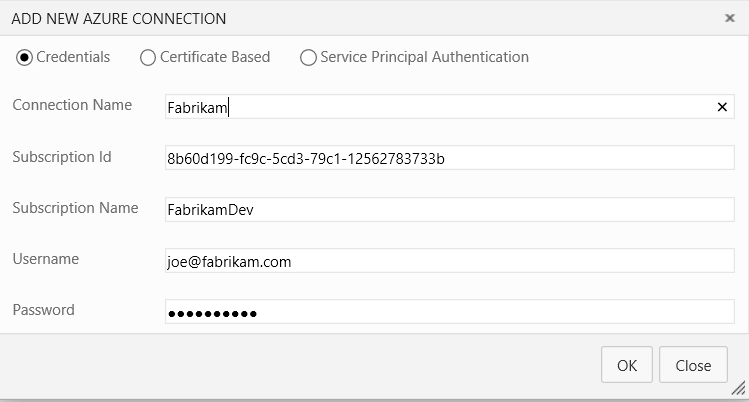
## Connect to the Azure Subscription

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

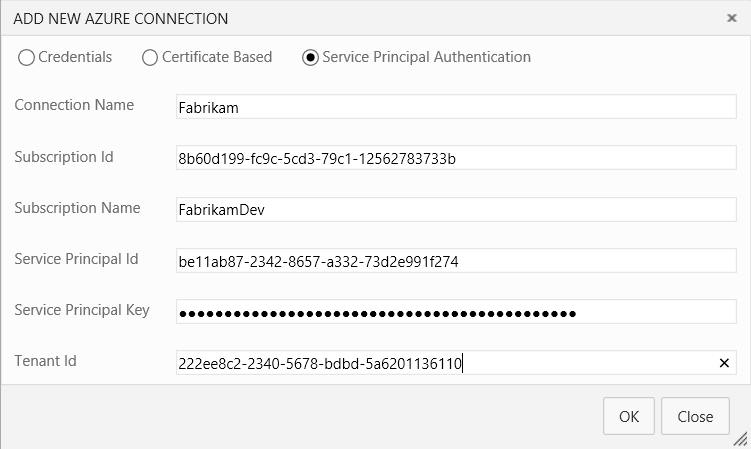
1. Select Credentials and enter a user friendly name for the connection name like Fabrikam.
2. Log into the [Azure portal](https://portal.azure.com/) and click on Subscriptions.
3. Copy the subscription ID and the subscription name and paste them into the Add Azure Subscription dialog box.
4. Enter a [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.

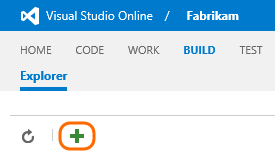


If there is no work account, then create a Service Principal as described [here](http://blogs.msdn.com/b/visualstudioalm/archive/2015/10/04/automating-azure-resource-group-deployment-using-a-service-principal-in-visual-studio-online-build-release-management.aspx) and enter the details as shown below:



## Create the 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

On the Build tab, add some steps.

|  |  |
| --- | --- |
|  | **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/testbed 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** | | adminUsername | Admin username of the VMs like vmuser or adminuser or any other | | adminPassword | Admin password of the VMs like Password123@~ or any other |   After entering the username and passwords click on the lock icon next to them. This will store them securely and they will not be visible to anyone else. |
|  | **Provision Virtual Machines in Azure using Templates**  Click on the Add build step and from the deployment tasks, select the Azure Resource Group Deployment task. Select the Azure subscription from the dropdown and fill in the other parameters of the task as described below:  **Action**  Select Create or Update Resource Group.  **Resource Group**  Enter the name of a Resource Group that does not exist in the subscription like, FabrikamTestbed1. If the Resource Group already exists in the Subscription, then it will be updated with the resources specified in the template.  **Location**  Select the location of the Azure datacenter where the Resource Group will be deployed like, East US. Note that if this Resource Group already exists in the subscription, then this parameter will be ignored.  **Template**  Use the Source Control picker to select the template file and the path will AzureRGTemplates/AzureSimpleVM.json  **Template parameters**  Use the Source Control picker to select the template parameters file and the path will be AzureRGTemplates/AzureSimpleVM.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 zrewsa123 can be changed to rezwsa321.  -newStorageAccountName zrewsa123 -vmName FabTstxyz123 -vmLocation "East US" -adminUsername $(adminUsername) -adminPassword $(adminPassword) -dnsNameForPublicIP fabtest1234 |

## Finish and test the definition

1. Save your definition and queue the build using the default Hosted build agent.
2. After a successful build, check your Resource Groups in the [Azure portal](https://portal.azure.com/).