WorkshopPLUS

Microsoft Azure Service Fabric for Developers

Create a Service Fabric Secure Cluster

Student Lab Manual

V1.3, March 15, 2017

Information in this document is subject to change without notice. The example companies, organizations, products, people, and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred. 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.

Microsoft may have patents, patent applications, trademarked, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any 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.

© 2010 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, MS, Windows, Windows NT, MSDN, Active Directory, BizTalk, SQL Server, SharePoint, Outlook, PowerPoint, FrontPage, Visual Basic, Visual C++, Visual J++, Visual InterDev, Visual SourceSafe, Visual C#, Visual J#, and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation in the U.S.A. and/or other countries.

Other product and company names herein may be the trademarks of their respective owners.

Contents

[Create a Service Fabric secure cluster 4](#_Toc477340091)

[Introduction 4](#_Toc477340092)

[Objectives 4](#_Toc477340093)

[Prerequisites 4](#_Toc477340094)

[Exercise 1 5](#_Toc477340095)

[Task 1 – Setting up your cluster certificate 5](#_Toc477340096)

[Task 2 – Setting up Azure Active Directory 6](#_Toc477340097)

[Task 3 – Create your Service Fabric cluster 10](#_Toc477340098)

[Task 3 – Test your Admin and Read-only user access 15](#_Toc477340099)

# Create a Service Fabric secure cluster

## Introduction

Estimated time to complete this lab

60 minutes

### Objectives

In this hands-on lab, you will learn how to:

* Create a self-signed certificate and install it in Azure Key Vault
* Set up of Azure Active Directory to use Role Based Access to the cluster
* Secure cluster creation
* Test your Admin and Read-only user login

## Prerequisites

The following is required to complete this hands-on lab:

* Microsoft Azure PowerShell
* [Microsoft Azure SDK for .NET for Visual Studio 2017](http://www.microsoft.com/windowsazure/sdk/)
* Microsoft Azure Service Fabric SDK – 2.4.164
* A Microsoft Azure subscription

# Exercise 1

In this exercise, you will learn how to setup a Service Fabric secure cluster that uses Azure Active Directory to implement role based access control to allow both administrative and read-only access control to the cluster and the Service Fabric Explorer.

## Task 1 – Setting up your cluster certificate

In this task, you will be executing a PowerShell script to create a Microsoft Azure KeyVault and a self-signed certificate. The script will add the certificate to the Azure KeyVault. **Do not close** the PowerShell window after you have executed the script. You will need the **Resource ID**, **Secret Url** and **Thumbprint** output values later in this lab.

1. On your machine, created a sub-directory where the upcoming execution of the PowerShell script can place your self-signed certificate. Example C:\Certs.
2. Open PowerShell ISE if it is not currently open. Execute the following command from the PowerShell command window to log in to your Azure account.

Login-AzureRmAccount

1. From the output in the PowerShell command window, copy the ***SubscriptionId*** and ***TenantId*** in to a text editor.
2. Open the file **.\Assets\CreateVaultCerts.ps1** and view it in the PowerShell ISE script pane window (CTRL-R if the window isn’t open). You will need to change the names of all the variables to match what you want to see in your own environment. Once you have changed the variable names, save the script and then set a breakpoint within the script code and select **F5** to start debugging. Step through the execution of the PowerShell script.

These are the variables you have to change in the PowerShell script.

#-----You have to change the variable values-------#

#Name of the Key Vault service

$KeyVaultName = "vaultname"

#Resource group for the Key-Vault service.

$ResourceGroup = "vaultrg"

#Set the Subscription

$subscriptionId = "subscriptionId"

#Azure data center locations (East US", "West US" etc)

$Location = "location"

#Password for the certificate

$Password = "password"

#DNS name for the certificate

$CertDNSName = "name.domain.com"

#Name of the secret in key vault

$KeyVaultSecretName = "servicecert"

The information you need to record from the PowerShell command prompt window will appear similar to this. Copy and paste this information in to a text editor:

Resource Id: /subscriptions/<your-subscriptionID>/resourceGroups/<your-resource-group>/providers/Microsoft.KeyVault/vaults/<your-vault-name>

Secret URL : https://<your-vault-name>.vault.azure.net:443/secrets/<secret>/<generated-guid>

Thumbprint : <certificate-thumbprint>

NOTE: When the Resource Id is copied from the PowerShell command prompt window, you will notice that the resource id is ‘wrapped’ and does not appear to be a single line of text. You need to make sure you remove the wrap from this line, otherwise your resource id will not be correct when it is used later.

## Task 2 – Setting up Azure Active Directory

To secure the cluster with Azure AD, you will need to decide which AD directory in your subscription you will be using. In this example, we will use the 'default' directory. In the previous task when you logged in to Azure with Login-AzureRmAccount, you should have recorded the 'tenantID'. This is the ID associated with your default AD directory. NOTE: If you have more than one directory (or tenant) in your subscription, you are going to have to make sure you get the right tenantID from your AAD administrator.

1. Within PowerShell ISE, open **.\Assets\SetupApplications.ps1**. You will not need to change any variables in this script file, but you can read through it to get an understanding of what is being accomplished. This script creates an AD application that will be registered with your Azure Active Directory tenant and the cluster will use this application to setup role based access control for your admin and read-only user.
2. In the PowerShell command prompt window, change the directory to your **.\Assets** directory.
3. Execute the following command from the PS command prompt window:

.\SetupApplications.ps1 -TenantId '<your-tenantID>' -ClusterName '<your-cluster-name>.<region>.cloudapp.azure.com' -WebApplicationReplyUrl 'https://<your-cluster-name>.<region>.cloudapp.azure.com:19080/Explorer/index.html'

Parameter descriptions:

* The **ClusterName** is used to prefix the AAD applications created by the script. It does not need to match the actual cluster name exactly as it is only intended to make it easier for you to map AAD artifacts to the Service Fabric cluster that they're being used with. This can be a bit confusing because you haven't created your cluster yet. But, if you know what name you plan to give your cluster, you can use it here.
* The **WebApplicationReplyUrl** is the default endpoint that AAD returns to your users after completing the sign-in process. You should set this to the Service Fabric Explorer endpoint for your cluster, which by default is: https://<cluster\_domain>:19080/Explorer

NOTE: You may be prompted to agree to run the script and to log in to Azure. If you are, select the Run once button, login to Azure, or accept whatever request you receive in order to execute the script. You may see some warnings about missing .dlls in the output window, you can ignore these warnings.

For a full list of AAD helper scripts, you can find more of these at <http://servicefabricsdkstorage.blob.core.windows.net/publicrelease/MicrosoftAzureServiceFabric-AADHelpers.zip>.

1. Record the information at the bottom of the command prompt window. You will need this information when you deploy your cluster from the portal. The information will look similar to what you see below.

"azureActiveDirectory": {

"tenantId":"<Your-AAD-tenantID>",

"clusterApplication":"1xxxxxxxx-x68e-490a-89c8-2894e4b8686a",

"clientApplication":"xxxxxxx-7825-4e1e-a586-f0ff8d9e679e"

1. Log in to the classic Azure portal at <https://manage.windowsazure.com>. For now, you need to use the classic Azure portal because the production portal Azure Active Directory features are still in preview.
2. Find the Active Directory menu item and click on it.



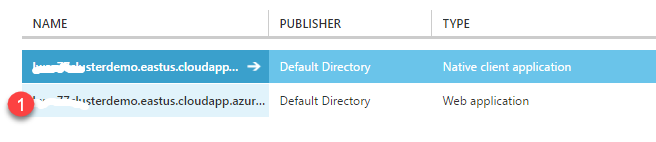
1. Find your Azure Active Directory in the list and click on it.

[](file:///C:\Users\larrywa\AppData\Local\Temp\WindowsLiveWriter1286139640\supfiles1720121\clip_image0014.jpg)

1. Click on the **Users** menu item at the top of the Active Directory window.
2. Add 2 new users to your directory. Name them whatever you want just as long as you know which one is Admin and which one would be the read-only user. Make sure to record the password that is initially generated, because the first time you try to log in to the portal as this user, you will be asked to change the password.
3. Within your AAD, click on the **Applications** menu. In the **Show** drop-down box, pick **Applications My Company Owns** and then click on the check button over to the right to do a search.



1. You should see two applications listed. One will be for Native client applications and the other for Web Applications. Click on the application name for the web application type. Since we will be doing our connectivity test connecting to the Service Fabric Explorer web UI, this is the application we need to set the user permissions on.



1. Click on the **Users** menu.
2. Click on the user name that should be the administrator and then select the **Assign** button at the bottom of the portal Window.
3. In the Assign Users dialog box, pick **Admin** from the dropdown box and select the check button.

[](file:///C:\Users\larrywa\AppData\Local\Temp\WindowsLiveWriter1286139640\supfiles1720121\clip_image0044.jpg)

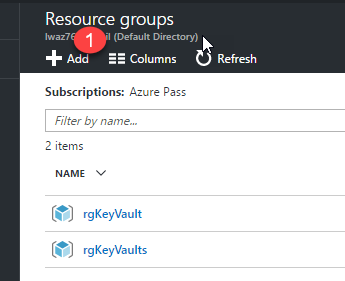
1. Repeat 13 and 14, but this time, for the read-only user select Read-only from the Assign Users drop-down. This step completes what you will need to do in the classic portal, so you can close the classic portal window.

You now have all the information you need to create your cluster in the portal. Log in to the Azure Portal at <https://portal.azure.com>.

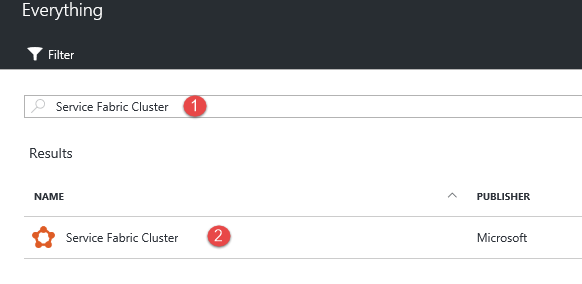
NOTE: If you intend on deploying a Service Fabric application from Visual Studio to this cluster, you will need to assign the user who will be logging in to Azure from Visual Studio to be an administrator also. When you go to deploy a Service Fabric application from Visual Studio, it will always ask you to log in to Azure. The admin and read-only user you just created do not have permissions to log in to an Azure subscription, they only have permissions to get to the Service Fabric cluster.

## Task 3 – Create your Service Fabric cluster

1. Log in to the Azure portal at <https://portal.azure.com>.
2. Click on the **Resource Groups** menu item on the left and then click the **+Add** button to create a new resource group for your Service Fabric cluster. It is not recommended to put your cluster in to the same resource group that your key vaults are located in. You may want to use the key vaults resource group for ONLY key vault information.



1. Once you have created your new resource group, click on the **Add** button in the resource group blade and in the **Everything** blade, type ‘*service fabric’* into the search field. You will be shown an option to select the **Service Fabric Cluster** item. Select it.



1. Select the **Create** button. When you do this, a set of blades will open that will allow you to start entering information to build your cluster. This is where you will need information that was displayed in the PowerShell command window from the previously executed PowerShell script.
2. In the **Basics** blade, fill in the following information and then select the **Ok** button.

|  |  |
| --- | --- |
| **Cluster Name** – a unique name (in all of Azure) of your Service Fabrc cluster.  **User name** – RDP user name for the machines in your cluster.  **Password** – RDP password for the machines in your cluster. Make sure this password is at least 12 characters long.  **Resource Group** – the name of the resource group to put the cluster in (choose the same resource group the keyvault is in)  **Location** – location (region) where you want the cluster to reside. | C:\Users\LARRYL~1\AppData\Local\Temp\SNAGHTML5130d11.PNG |

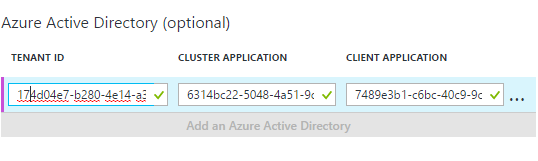
1. In the **Cluster Configuration** blade, enter the following information and then select the **Ok** button when you have the information filled in.

|  |  |
| --- | --- |
| **Node type count** = 1  Node type -…  **Node type name** – name of your nodes that will display in the Service Fabric explorer.  **Durability tier** – Bronze  **Virtual machine size** – you can choose an appropriate size (larger machines cost more money.)  **Reliability tier** - Silver  **Custom Endpoints** – enter any port number here that you need to have exposed to a load balancer. You can enter multiple, separated by commas. You can also enter this information after the cluster has been created. |  |

1. In the **Security** blade, the first information you need to fill out will be the information that you recorded in Task 1, step 4 (resource ID etc). Go to your text editor where you pasted this information in and enter it below, but **DO NOT** select the OK button yet.

|  |  |
| --- | --- |
| **Security mode** – leave this set to Secure  **Source key vault** – this is actually the ‘Resource Id’ value from the PowerShell script  **Certificate URL** – this is the ‘secret URL’ value from the PowerShell script  **Certificate thumbprint** – from the PowerShell script value |  |

1. Select the **Configure advanced settings** checkbox (check it).
2. Scroll down in the blade until you see the **Azure Active Directory (optional)** section.
3. Click on the  button.
4. Fill in the information below based on the information you recorded in Task 2, step 4 and then select the **OK** button.

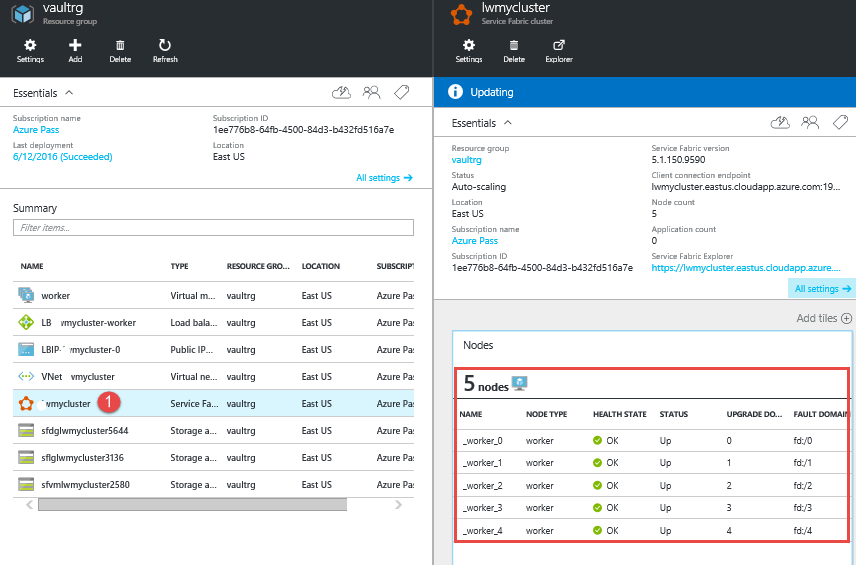


**Tenant ID** – your Active Directory tenant ID

**Cluster application** – the web application GUID from your AD tenant

**Client application** – the native application GUID from your AD tenant

1. In the **Summary** blade, select the **Create** button. The creation of the cluster may take up to 20 minutes. Note that even when it appears that the cluster has completed creation, you will still need to wait for the cluster to create the nodes. The screenshot below shows nodes created.



## Task 3 – Test your Admin and Read-only user access

1. Once the cluster has completed the creation process, make sure you log out of the Azure portal. This assures that when you attempt to log in as the Admin or Read-only user, you will not accidentally log in to the portal as the subscription administrator.
2. Log in to the portal as either the Admin or Read-only user. You will need to change the temporary password you were provided early and then the log in will complete.
3. Open up a new browser window and log in to [https://<yourfullclustername>:19080/Explorer/](https://msdnshared.blob.core.windows.net/media/2016/12/Setting-up-a-Service-Fabric-Cluster-secured-by-Azure-AD1.docx). Test the Explorer functionality. For true testing, you may need to deploy a Service Fabric application to this portal first. For this, you could go back to the lab exercise in the Introduction to Service Fabric module and deploy that application.

NOTE: When logging in to the Service Fabric Explorer as a read-only user, the menus for deactivating services, nodes etc. will not be disabled. But, if you try to perform those operations, you will see an access denied message.