

Azure Storage Solutions

- Azure Storage Labs



May 12, 2017

Version 1.2 Final

Prepared by

Niko Dukic
Global Black Belt TSP – Hybrid Storage CEE
niko.dukic@microsoft.com

Contributors

Document Revision

Change Record

Date	Author	Version	Change Reference
04/05/2017	Niko Dukic	1.1	Added StorSimple Labs
12/05/2017	Patrick Mandemaker	1.2	Combined different versions

Reviewers

Name	Version Approved	Position	Date
Yann Seyroles			02/23/2017
Patrick Mandemaker			03/30/2017
Benedict Berger			02/23/2017

Table of Contents

Document Revision	2
Change Record	2
Reviewers	2
Introduction.....	5
Objectives	5
Note regarding pre-release software	6
Note regarding user account control	6
Note regarding network access control.....	6
Activity 1: Getting Started with Azure Storage.....	7
Objectives	7
Exercise 1: Create a new resource group	8
Exercise 2: Create a new storage account	12
Activity 2: Using Storage Account Blob Service.....	15
Objectives	15
Exercise 1: Working with Storage Account blob service	16
Create a new container	16
Access to storage account objects	19
Exercise 2: Using SAS Tokens to access storage account objects	21
Revoke SAS Token.....	25
Using SAS tokens with stored access policy	26
Change SAS Token with Stored Access Policy	32
Activity 3: Using Storage Account File Service.....	34
Objectives	34
Exercise 1: Working with Azure Files	35
Create a new file share.....	35
Connect to file share.....	36
Activity 4: Using StorSimple Virtual Array.....	40
Objectives	40
Exercise 1: Working with StorSimple	41
Create a StorSimple service manager	41

Provision a StorSimple Virtual Array.....42

Register StorSimple Virtual Array43

Configure StorSimple Virtual Array.....45

Provision a Windows host47

Add a new volume.....48

Mount a volume to the windows host.....49

Create a cloud backup50

Restore the volume51

Failover to a new device.....52

Introduction

Azure Storage is a foundational building block that is core to most of the other Microsoft cloud service. It represents the basic service that supports all Azure services like virtual machines, HDInsight, Azure SQL but also other Microsoft services like Xbox and O365 services.

Objectives

During this lab, you will learn the basics of Azure Storage Accounts:

- Create a general-purpose storage account
- Create a blob container
- How to use Storage Explorer to manage storage account access
- Create a SAS access token and access the storage account objects
- Create a SAS Access Policy and access the storage account objects
- Create a file share
- Mount a file share to an on-premises client or an Azure VM

Estimated time to complete this lab

120 minutes

Prerequisites

- Access to Azure Portal with either your own Azure Subscription or Azure Pass. If you need instructions how to create Microsoft ID and get Azure Pass to access Azure portal, you can read detailed instructions on Exercise 1 and Exercise 2 of Activity 1 in OMS Log Analytics Lab. Instructions will guide you how to create Microsoft ID, get a promo code for Azure Pass that will allow you to access all necessary services for this lab.
- Installation of Microsoft Azure Storage Explorer tool from <http://storageexplorer.com/>

Additional resources

Note regarding pre-release software

Portions of this lab may include software that is not yet released, and as such may still contain active or known issues. While every effort has been made to ensure this lab functions as written, unknown or unanticipated results may be encountered because of using pre-release software.

Note regarding user account control

Some steps in this lab may be subject to user account control. User account control is a technology which provides additional security to computers by requesting that users confirm actions that require administrative rights. Tasks that generate a user account control confirmation are denoted using a shield icon. If you encounter a shield icon, confirm your action by selecting the appropriate button in the dialog box that is presented.

Note regarding network access control

Some steps in this lab may be subject to network access control limited by your company, venue where the labs are organized or an ISP. We will note networking requirements for the labs but we will not be able to control them. In case any error occurs, one way how to accomplish the labs is to provision a virtual machine in Azure and run the labs from there instead of your own client.

Activity 1: Getting Started with Azure Storage

Estimated time to complete this activity

30 minutes

Objectives

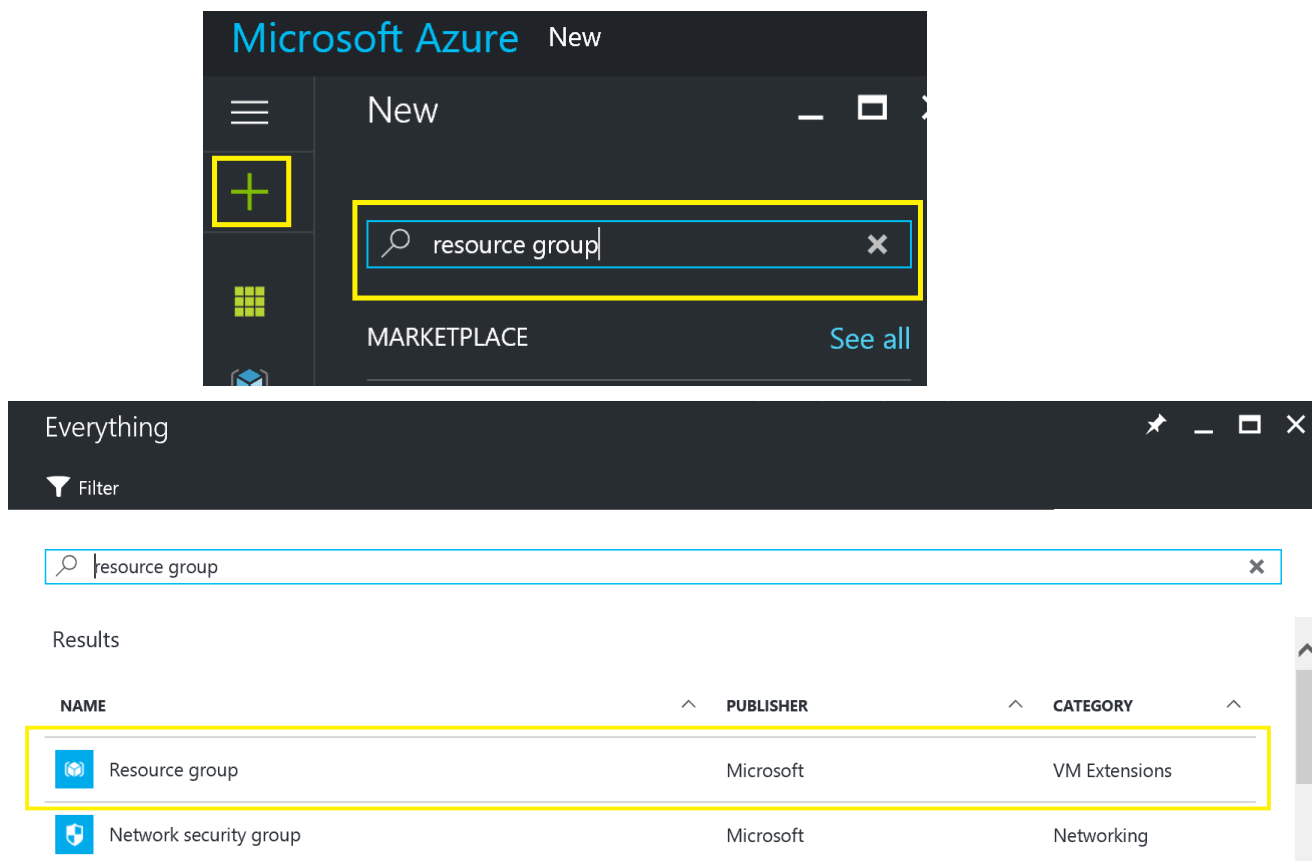
In this activity, you will create a general-purpose storage account and walk through the basic functionalities.

- ✦ You can use an existing storage account, but it's recommended to use a new one.
- ✦ We also strongly recommend that you use InPrivate browsing to ensure that you are not automatically logged on with other credentials during the registration / activation process.

Exercise 1: Create a new resource group

In this task, you will create a new resource group.

1. On the Internet Explorer Favorites Bar, click **Azure Portal**.
★ The Microsoft Azure Portal will open from <http://portal.azure.com>.
2. Sign in with the credentials you created in the prerequisites.
3. In Azure Portal search for **Resource Group** (Click on the '+' sign or press 'N' and type 'resource group').
4. Click on **Resource group** (Publisher Microsoft).
5. Click **Create**
Name: **StorageLabs**
Subscription: **the one created in prerequisites (Azure Pass)**
Location: **West Europe**
6. Check '**Pin to dashboard**' for easy access in following exercises
7. Click **Create**





Resource group

Microsoft



Resource groups enable you to manage all your resources in an application together. Resource groups are enabled by Azure Resource Manager. Resource Manager allows you to group multiple resources as a logical group which serves as the lifecycle boundary for every resource contained within it. Typically a group will contain resources related to a specific application. For example, a group may contain a Website resource that hosts your public website, a SQL Database that stores relational data used by the site, and a Storage Account that stores non-relational assets.



PUBLISHER

Microsoft

USEFUL LINKS

[Documentation](#)

Create

Resource group

Create an empty resource group

* Resource group name

StorageLabs

* Subscription

Azure Pass

* Resource group location

West Europe

☒ Pin to dashboard

Create

Wait till resource group is created.

You can monitor the resource group creation by clicking on the notification button. This applies to any activity you make on the portal.

×

🔔

⚙️

😊

?

gbblabs@outlook.com

GBBLABSOUTLOOK (DEFAULT...

👤

×

Notifications

Dismiss all

✓

Creating resource group 'StorageLabs' su... 12:42

Creating resource group 'StorageLabs' was successful.

Exercise 2: Create a new storage account

In this task, you will create a new general-purpose storage account

1. In Azure Portal search for Storage Account (Click on the '+' sign or press 'N' and type '**storage account**').
2. Click on **Storage Account** (Publisher Microsoft).
3. Click **Create**

Name: **unique name** across azure storage services (We will use the name storagelabs but please enter a random name here, only small letters. If the name is taken change the name until you see a green check mark next to the name as shown on the screen).

Deployment model: **Resource Manager**

Account kind: **General purpose**

Performance: **Standard**

Replication: **Locally-redundant storage (LRS)**

Storage service encryption: **Disabled**

Subscription: **the one created in prerequisites (Azure Pass)**

Resource Group: **Use Existing -> StorageLabs** (if you don't see resource group created in the exercise 1, please refresh the browser and try again.)

Location: **West Europe**

4. Check '**Pin to dashboard**' for easy access in following exercises
5. Click **Create**

Microsoft Azure New > Create storage account

Create storage account

usage and the options you choose below.
[Learn more](#)

* Name ⓘ
storagelabs ✓

Deployment model ⓘ
Resource manager Classic

Account kind ⓘ
General purpose

Performance ⓘ
Standard Premium

Replication ⓘ
Locally-redundant storage (LRS)

* Storage service encryption ⓘ
Disabled Enabled

* Subscription ⓘ
Azure Pass

* Resource group ⓘ
☐ Create new ☒ Use existing
StorageLabs

* Location ⓘ
West Europe

☒ Pin to dashboard

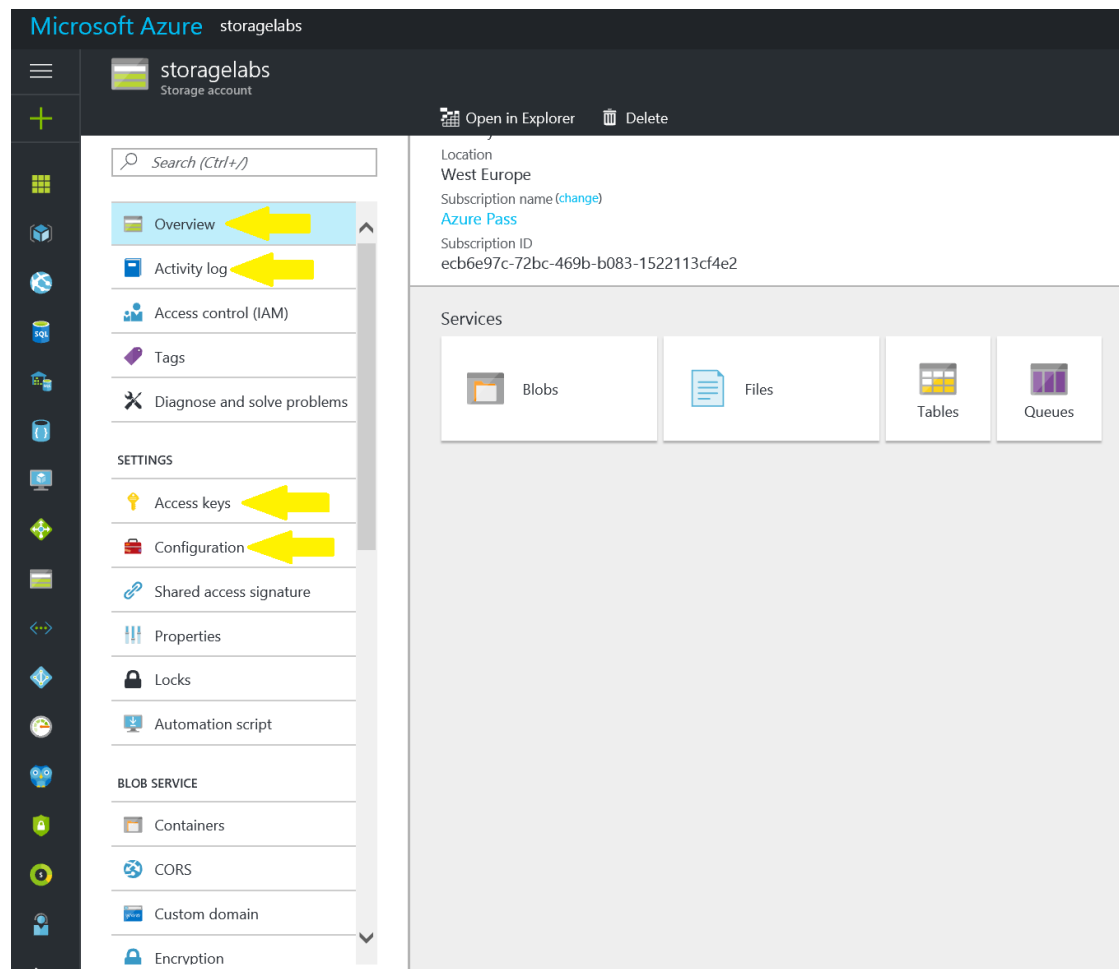
Create [Automation options](#)

Wait till the storage account gets created, it will automatically redirect you to the storage account overview windows.

Tip: if you want to use powershell for creation of Storage Accounts, you can use the command [Get-AzureRmStorageAccountNameAvailability](#) to check if name is available.

In this task, get familiar with storage account settings and functionalities for easier navigation.

1. Find the storage account created in Activity 1 / Exercise 2 and open Overview window (Storage Account should have been placed on main dashboard if 'Pin to dashboard' was selected, if not use search in Azure portal and type the name you defined in exercise 2).
2. Get familiar with storage account portal. Walk through following menus on the left:
 - a. **Overview**: shows main Storage Account services as well as basic monitoring metrics
 - b. **Activity log**: shows all the activities on the storage account. You should see when the storage account was created and by whom
 - c. **Access Keys**: shows main security keys for Storage Account access (key 1 and key 2). This keys will be needed in the following exercises so note where you can find them
 - d. **Configuration**: Shows storage account performance and replication settings. Performance can't be changed after creation but you can change replication on the fly.



Activity 2: Using Storage Account Blob Service

Estimated time to complete this activity

75 minutes

Objectives

In this activity, you will upload some objects to storage account blob service and apply different access methods to access them securely.

- ✦ You can use any existing general-purpose storage account, but it's recommended to use the one created in Activity 1
- ✦ We also strongly recommend that you use InPrivate browsing to ensure that you are not automatically logged on with other credentials during the registration / activation process.

Exercise 1: Working with Storage Account blob service

In this exercise, you will learn how to create a new container and upload files to the container and basics of access to storage account objects.

Create a new container

In this task, we will create a new container in blob service of the storage account created in Activity 1.

1. Find the storage account created in Activity 1 / Exercise 2 and open Overview window (Storage Account should have been placed on main dashboard if 'Pin to dashboard' was selected, if not use search in Azure portal and type the name you defined in Activity 1 / Exercise 2).
2. From the list of services on the Overview window select '**Blob**'
3. On the next window, click '+ **Container**' to add a new container to the selected blob service
Name: **Container1** (name can contain only small letters and numbers and must be unique on a storage account level)
Access type: **Container**
4. Click **Create**

Note: There are 3 different Access types when you create a new container. Please read them to better understand next exercises with building security mechanism for blob access.

- Private: Container is accessible only by storage account owner. Access needs any of the security mechanism like access keys or SAS tokens.
- Container: Container and blob data can be read via anonymous request. Clients can enumerate blobs within the container via anonymous request, but cannot enumerate containers within the storage account
- Blob: Blob data within this container can be read via anonymous request, but container data is not available. Clients cannot enumerate blobs within the container via anonymous request

Blob service
storagelabs

+ Container

Refresh

Essentials

Search containers by prefix

NAME	URL	LAST MODIFIED
No containers found.		

New container
Blob service (storagelabs)

* Name

container1

Access type

Container

Create

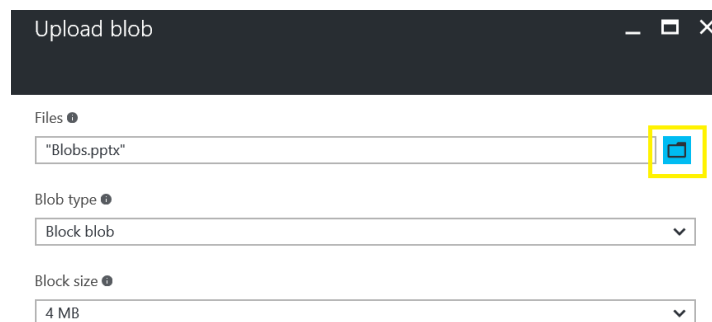
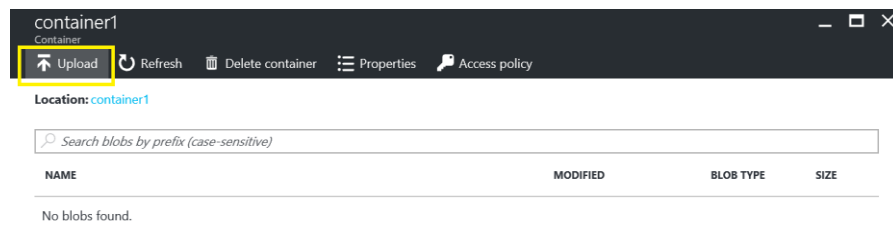
In this task, you will upload a file to the newly created container in storage account.

1. Click on the newly created container **container1**.
2. Select the option **Upload**
3. Select any file to upload via browse button

Blob type: **Block blob**

Block size: **4MB**

4. Click **Upload**



Wait till the upload process is finished.

You can monitor the upload process by clicking on the notification tab.

Tip: If you need to store a blob in the root folder of the storage account, you can create a special container with the name \$root and store the blobs there.

Access to storage account objects

In this task, you will learn how to access the data stored in a blob service and how to apply different access policies. To access data in storage account we will use any browser that can access the data on a public internet via http and https. We will access the data by entering couple of different addresses to understand different access types. You can copy the links to the storage account by browsing to the uploaded file in azure portal from the previous task.

1. Open a browser of your choice
2. Enter the following address in the address bar where <storage account name> is the storage account name created in Activity 1 / Exercise 2, while <uploaded file> is file uploaded in Activity 2 / Exercise 1.

`http://<storage account name>.blob.core.windows.net/container1/<uploaded file>`

(in our lab example full link is

<https://storagelabs.blob.core.windows.net/container1/Blobs.pptx>).

3. Enter the following address in the address bar where <storage account name> is the storage account name created in Activity 1 / Exercise 2. Note the added string &comp=list&restype=container at the end of URL.

`http://<storage account`

`name>.blob.core.windows.net/container1?comp=list&restype=container`

(in our lab example full link is

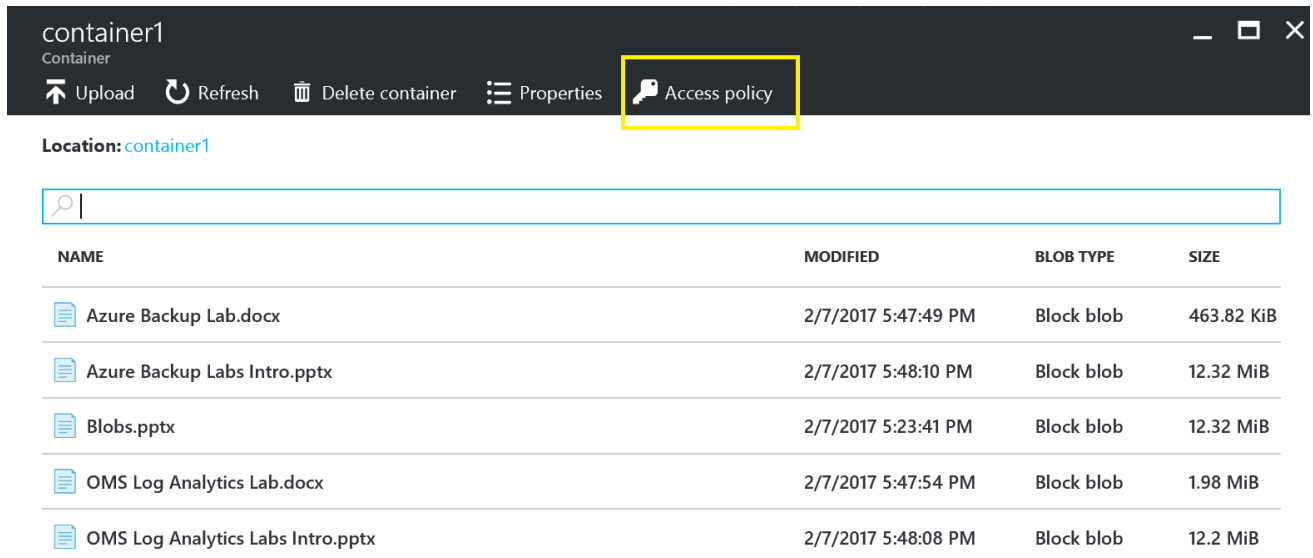
<https://storagelabs.blob.core.windows.net/container1?comp=list&restype=container>).






Explanation: In step 2 we can access the uploaded file since we selected 'Container' access policy during container creation. This allows us to read any blob stored on that container as well as to enumerate the blobs that are stored inside that blob container. You can try to upload some more files and run the address from the step 3 again. You should see all the uploaded files with its properties as well as read all the files with the direct link you can find in <Url> parameter of the xml output.

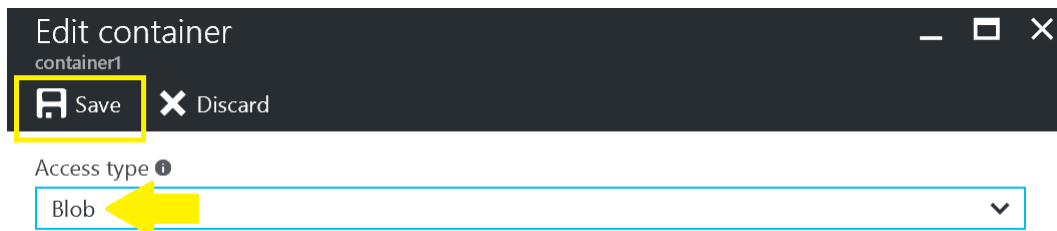
Note: by default, http and https are enabled on the storage account. Repeat steps 2 and 3 with http protocol instead of https protocol and they will still work. We are strongly encouraging not to use http. This behavior can be enforced by SAS tokens which we will show in the later exercises.

In this task, you will change the access type to 'Blob' and test how this affect access to the stored blobs and container.

1. Find the storage account created in Activity 1 / Exercise 2 and open Overview window (Storage Account should have been placed on main dashboard if 'Pin to dashboard' was selected, if not use search in Azure portal end type the name you defined in Activity 1 / Exercise 2).
2. From the list of services on the Overview window select '**Blob**'
3. From the list of containers select container created in Activity 2 / Exercise 1 named **container1**
4. Click on **Access policy**
5. Change **Access Type** to **Blob** and click **Save**



NAME	MODIFIED	BLOB TYPE	SIZE
 Azure Backup Lab.docx	2/7/2017 5:47:49 PM	Block blob	463.82 KiB
 Azure Backup Labs Intro.pptx	2/7/2017 5:48:10 PM	Block blob	12.32 MiB
 Blobs.pptx	2/7/2017 5:23:41 PM	Block blob	12.32 MiB
 OMS Log Analytics Lab.docx	2/7/2017 5:47:54 PM	Block blob	1.98 MiB
 OMS Log Analytics Labs Intro.pptx	2/7/2017 5:48:08 PM	Block blob	12.2 MiB



Access type ⓘ

Blob

6. Open a browser of your choice
7. Enter the following address in the address bar where <storage account name> is the storage account name created in Activity 1 / Exercise 2, while <uploaded file> is file uploaded in Activity 2 / Exercise 1.

http://<storage account name>.blob.core.windows.net/container1/<uploaded file>

(in our lab example full link is

<https://storagelabs.blob.core.windows.net/container1/Blobs.pptx>).

8. Enter the following address in the address bar where <storage account name> is the storage account name created in Activity 1 / Exercise 2. Note the added string &comp=list&restype=container at the end of URL.

http://<storage account

name>.blob.core.windows.net/container1?comp=list&restype=container

(in our lab example full link is

<https://storagelabs.blob.core.windows.net/container1?comp=list&restype=container>).

9. What is the difference now comparing to last access attempt?

Explanation: In first access task both access requests were successful. In this task, we could read the blob object but could not enumerate the container content as the container access type was set to 'Blob'.

In this task, you will change the access type to 'Private' and test how this affect access to the stored blobs and container.

1. Follow the steps 1 – 4 from the previous task
2. Change **Access Type** to **Private** and click **Save**
3. Follow the steps 6 – 8 from the previous task
4. What is the difference now comparing to last two access attempts?

Explanation: In this access attempts both requests were unsuccessful as the container access type was set to 'private' which allows the access to the blobs and container only to the requestor that has Storage Account security keys. To reduce the need of sharing access keys but still to be allow to securely give access to containers and blobs, we can use SAS tokens.

Exercise 2: Using SAS Tokens to access storage account objects

In this exercise, you will learn how to security mechanism to secure the content. We will use two tools in this section, Azure portal as well as Storage Explorer developed by Microsoft.

In this task, you will learn how to create a SAS token to allow temporary access to objects and blobs but keeping the access type to 'private'.

1. Find the storage account created in Activity 1 / Exercise 2 and open Overview window (Storage Account should have been placed on main dashboard if 'Pin to dashboard' was selected, if not use search in Azure portal and type the name you defined in Activity 1 / Exercise 2).
2. On the Storage Account settings menu, select **Shared access signature** option and set the option as follows:
Allowed services: **Blob**
Allowed resource types: **Container, Object**
Allowed permissions: **Read, List**
Start: delete the time part or set to any date in the past
End: change the date to any date in the future
Time zone: change the time zone to match your current time zone
Allowed protocols: **HTTPS Only**
Signing key: **key2**
3. Click **Generate SAS**
4. Scroll down and copy the **SAS Token** (note: if you remove the current window you can't get the same SAS token and you will need to create a new one so best is to copy it to a txt file as we will use it in multiple tasks.)

Allowed services ⓘ

☒ Blob ☐ File ☐ Queue ☐ Table

Allowed resource types ⓘ

☐ Service ☒ Container ☒ Object

Allowed permissions ⓘ

☒ Read ☐ Write ☐ Delete ☒ List ☐ Add ☐ Create ☐ Update ☐ Process

Start and expiry date/time ⓘ

Start

2017-02-09 12:00:00 AM

End

2017-02-12 12:00:00 AM

UTC +02:00

Allowed IP addresses ⓘ

for example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Allowed protocols ⓘ

☒ HTTPS only ☐ HTTPS and HTTP

Signing key ⓘ

key2

Generate SAS

SAS token ⓘ

?sv=2015-12-11&ss=b&srt=co&sp=rl&se=2017-02-11T22:00:00Z&st=2017-02-08T22:00:00.

Blob service SAS URL

https://storagelabs.blob.core.windows.net/?sv=2015-12-11&ss=b&srt=co&sp=rl&se=2017-02-11T22:00:00Z&st=2017-02-08T22:00:00.

In this task, you will use the generated SAS token to access the storage account object data. Before, proceeding, check that the 'Access type' on the container is set to 'Private'.

1. Open a browser of your choice
2. Enter the following address in the address bar where <storage account name> is the storage account name created in Activity 1 / Exercise 2, <uploaded file> is file uploaded in Activity 2 / Exercise 1 and <SAS Token> is SAS Token generated and copied in previous task.

http://<storage account name>.blob.core.windows.net/container1/<uploaded file> <SAS Token>

(in our lab example full link is

https://storagelabs.blob.core.windows.net/container1/Blobs.pptx?sv=2015-12-

11&ss=b&srt=co&sp=rl&se=2017-02-11T22:00:00Z&st=2017-02-08T22:00:00Z&spr=https&sig=bXQh6973O3jbXulPrfqQpobL87ggO%2FH4V85TlevPkE0%3D).

3. Enter the following address in the address bar where <storage account name> is the storage account name created in Activity 1 / Exercise 2 and <SAS Token> is SAS Token generated and copied in previous task. Note the added string &comp=list&restype=container at the end of URL.

http://<storage account name>.blob.core.windows.net/container1<SAS Token>&comp=list&restype=container

(in our lab example full link is

https://storagelabs.blob.core.windows.net/container1?sv=2015-12-11&ss=b&srt=co&sp=rl&se=2017-02-11T22:00:00Z&st=2017-02-08T22:00:00Z&spr=https&sig=bXQh6973O3jbXulPrfqQpobL87ggO%2FH4V85TlevPkE0%3D&comp=list&restype=container).

4. Test both links with http protocol now and see what happens.
5. Generate the new SAS token following steps 1 – 4 from previous task but uncheck also 'Container' from 'Allowed resource types' and uncheck 'List' from 'Allowed permissions'
6. Enter the following address in the address bar where <storage account name> is the storage account name created in Activity 1 / Exercise 2 and <SAS Token> is SAS Token generated and copied in previous task. Note the added string &comp=list&restype=container at the end of URL.

http://<storage account name>.blob.core.windows.net/container1<SAS Token>&comp=list&restype=container

(in our lab example full link is

https://storagelabs.blob.core.windows.net/container1?sv=2015-12-11&ss=b&srt=o&sp=r&se=2017-02-11T22:00:00Z&st=2017-02-08T22:00:00Z&spr=https&sig=Vxza1EW7X0%2FRTM0X5A4YIHBhWyNn6RA94gRtvdI9hx8%3D&comp=list&restype=container)

What is the result of this request now?

Explanation: In this task, we learned that with SAS tokens we can grant access to private resources but also be much more selective on type of access we want to grant. We can grant access on service, container and / or object level as well as define permissions like read and write.


Tip: Since there is no way to revoke generated SAS except changing the access keys, in case you are giving access to external parties, we recommend to use key1 for your company access and key2 for external access to generate SAS. This gives you the flexibility to change only the key2

and revoke the access to external parties but keeping the same SAS tokens for your internal access. In case you change key2 all SAS tokens generated with that key will be invalidated.

Revoke SAS Token

In this task, you will learn how to revoke the SAS token access.

1. Open a browser of your choice
2. Enter the following address in the address bar where <storage account name> is the storage account name created in Activity 1 / Exercise 2, <uploaded file> is file uploaded in Activity 2 / Exercise 1 and <SAS Token> is SAS Token generated and copied in previous task.
http://<storage account name>.blob.core.windows.net/container1/<uploaded file> <SAS Token>
(in our lab example full link is
https://storagelabs.blob.core.windows.net/container1/Blobs.pptx?sv=2015-12-11&ss=b&srt=co&sp=rl&se=2017-02-11T22:00:00Z&st=2017-02-08T22:00:00Z&spr=https&sig=bXQh6973O3jbXulPrfqQpobL87ggO%2FH4V85TlevPKE0%3D).
3. Make sure this link works
4. Find the storage account created in Activity 1 / Exercise 2 and open Overview window (Storage Account should have been placed on main dashboard if 'Pin to dashboard' was selected, if not use search in Azure portal and type the name you defined in Activity 1 / Exercise 2).
5. On the Storage Account settings menu, select **Access keys** option
6. Click **Regenerate** for **key2**
7. Confirm by selecting **Yes**
8. Once the key is regenerated, try to use link used in step 2. Does it work now?


storagelabs - Access keys
 Storage account

★
—
□
×

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

SETTINGS

Access keys

Configuration

Shared access signature

Properties

Locks

Automation script

Use access keys to authenticate your applications when making requests to this Azure storage account. Store your access keys securely - for example, using Azure Key Vault - and don't share them. We recommend regenerating your access keys regularly. You are provided two access keys so that you can maintain connections using one key while regenerating the other.

When you regenerate your access keys, you must update any Azure resources and applications that access this storage account to use the new keys. This action will not interrupt access to disks from your virtual machines. [Learn more](#)

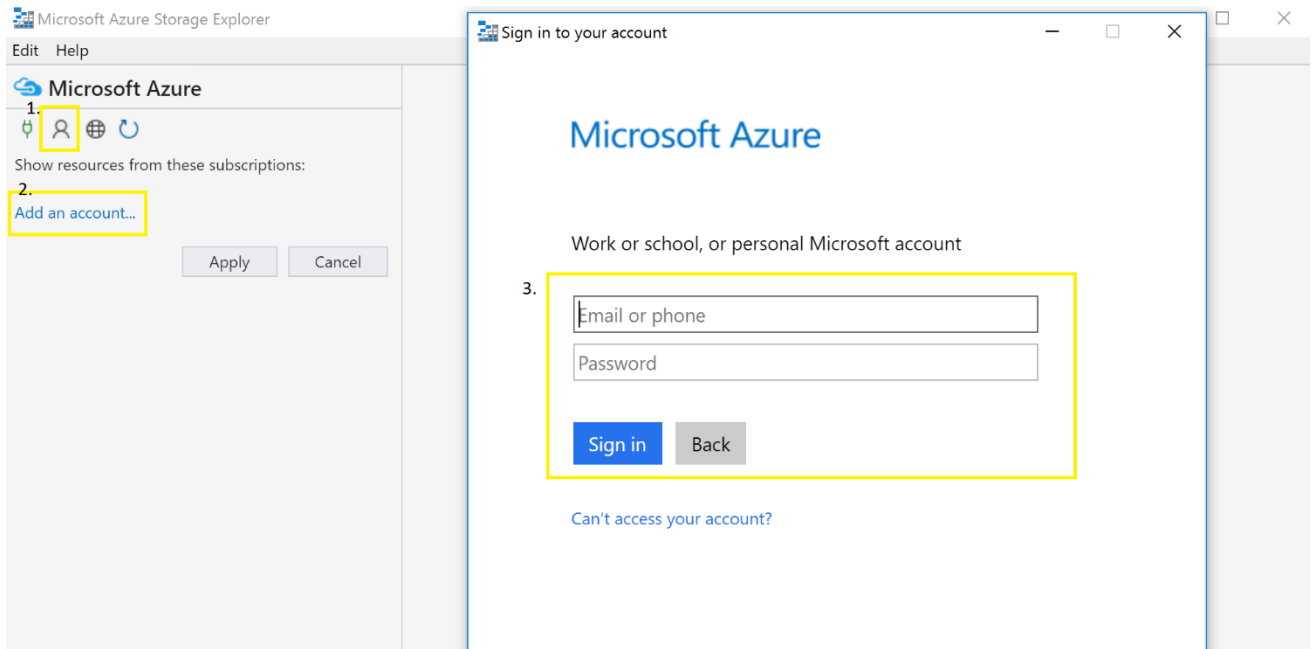
Storage account name

NAME	KEY			
key1	0sqrJsRyrrHS3QxLFrPCs0RtEGnypuedKO7lhKc9VeCC/HjMLkP			...
key2	pr3ehtYC2dOZbeTHs7QMQxMfcXQTOAU7LaiWeH9pNSTKEO			...

Using SAS tokens with stored access policy

In this task, you will learn how to allow access to objects and containers by using SAS tokens with stored access policy. Stored access policy allows more flexibility than pure ad hoc built SAS tokens as some of the parameters of the stored access policy can be changed (for example it can be revoked and / or extended without changing the access key). Currently, stored access policy can't be built using Azure Portal so we will use a tool called Microsoft Azure Storage Explorer (<http://storageexplorer.com/>). You can currently store up to 5 SAS policies per container.

1. Open Microsoft Azure Storage Explorer tool
2. Click on **Azure Account Settings**
3. Click on **Add an account**
4. Sign in with the credentials you created in the prerequisites

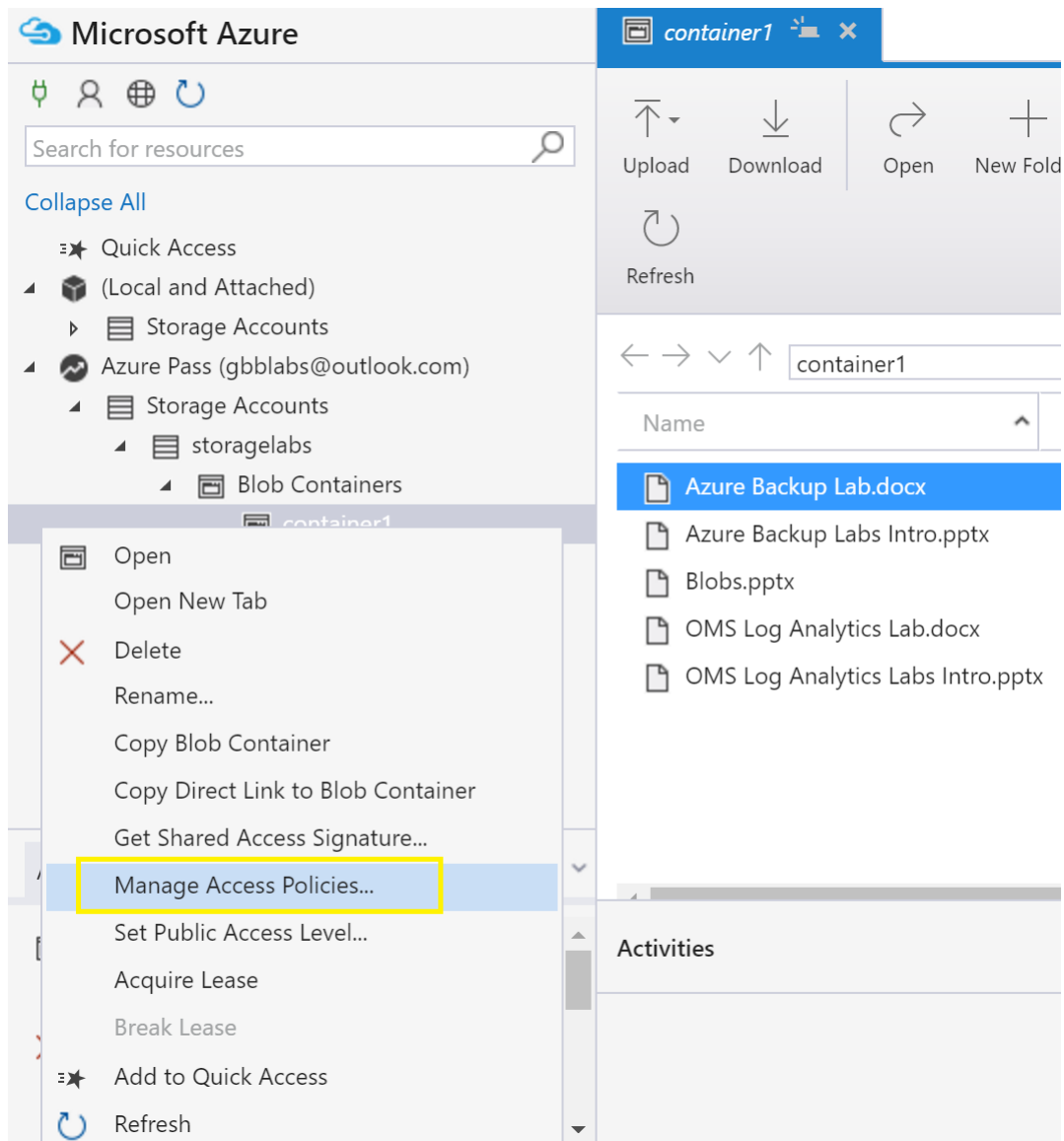


5. Click **Apply**
6. Expand your storage account and drill down to container level
7. Select container **container1**
8. Right-click **container1** and select **Manage access policies**
9. Click **Add**
Name: **SASPolicy1**
Start time: any date in the past
End time: any date in the future
Permissions: **Read, List**
Time zone: **Local**
10. Click **Save**
11. Right-click **container1** again and select **Get Shared Access Signature**
12. From Access policy drop-down menu select policy created in step 9 **SASPolicy1**
13. Click **Create**
14. Copy **Query string** (We recommend to copy the string in a text file)
15. Click **Close**
16. Open a browser of your choice
17. Enter the following address in the address bar where <storage account name> is the storage account name created in Activity 1 / Exercise 2 and <SAS Policy> is query string copied in step 14. Note the added string &comp=list&restype=container at the end of URL.
http://<storage account name>.blob.core.windows.net/container1<SAS Policy>&comp=list&restype=container

18. (in our lab example full link is

[https://storagelabs.blob.core.windows.net/container1?sv=2015-12-](https://storagelabs.blob.core.windows.net/container1?sv=2015-12-11&si=SASPolicy1&sr=c&sig=y4ldnLQdOgwkhp5T0j6d%2BmvW96wJkiNwRI0ZIFyn0ng%3D&comp=list&restype=container)

[11&si=SASPolicy1&sr=c&sig=y4ldnLQdOgwkhp5T0j6d%2BmvW96wJkiNwRI0ZIFyn0ng%3D&comp=list&restype=container](https://storagelabs.blob.core.windows.net/container1?sv=2015-12-11&si=SASPolicy1&sr=c&sig=y4ldnLQdOgwkhp5T0j6d%2BmvW96wJkiNwRI0ZIFyn0ng%3D&comp=list&restype=container)).



Access Policies

Container:

container1

Access policies:

Id	Start time	Expiry time	Read	Write	Delete	List	
SASPolicy1	01/09/2017 02:15 PM	03/16/2017 02:15 PM	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Remove

Add

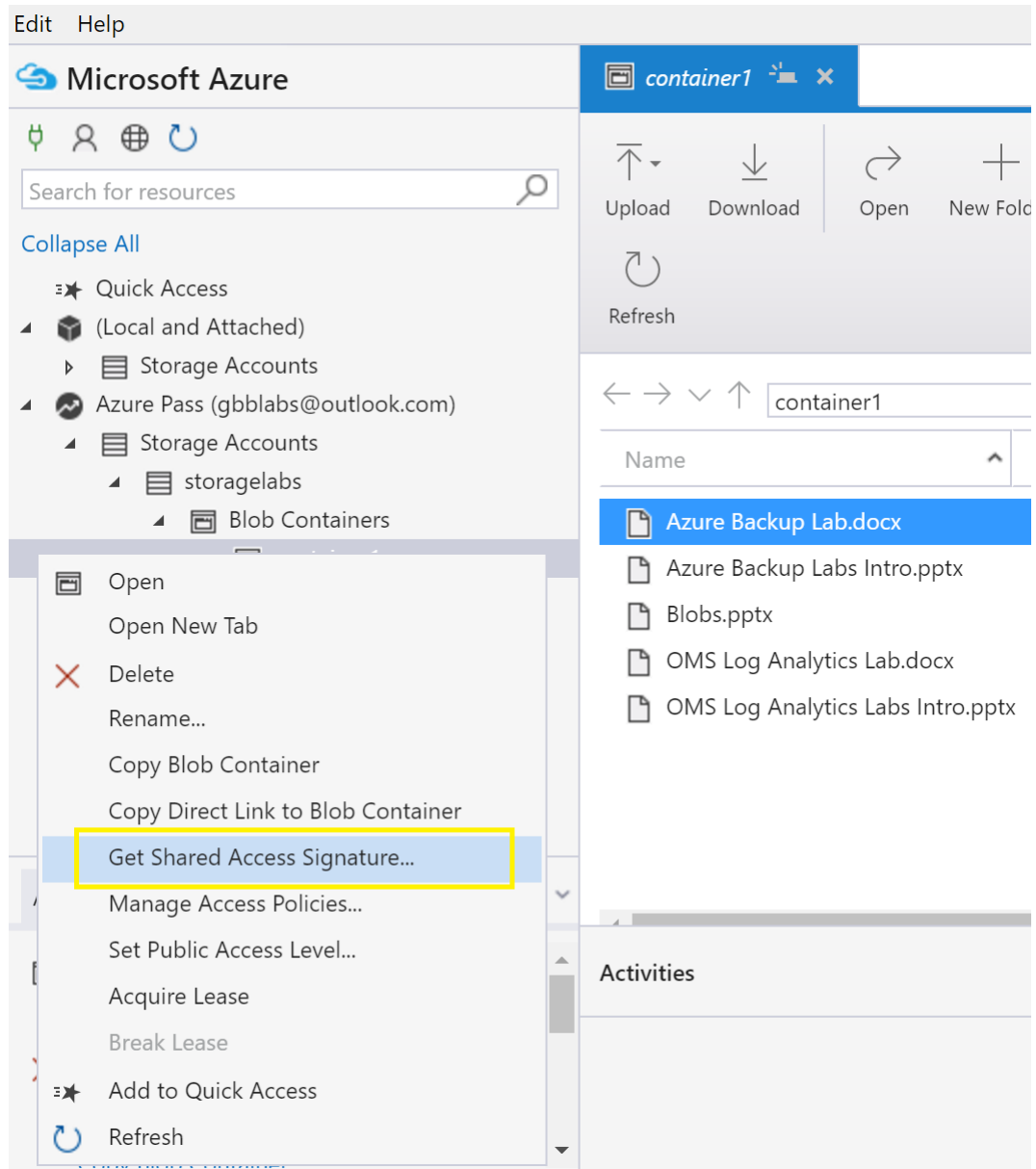
Time zone:

☒ Local

☐ UTC

Save

Cancel



Shared Access Signature

Access policy:

SASPolicy1



Start time:

01/09/2017 02:15 PM

Expiry time:

03/16/2017 02:15 PM

Time zone:

☒ Local

☐ UTC

Permissions:

- ☒ Read
- ☐ Write
- ☐ Delete
- ☒ List

Create

Cancel

Shared Access Signature

Container:

URL:

Query string:

Change SAS Token with Stored Access Policy

In this task, you will learn why using SAS Access Policy is more convenient than using ad hoc SAS token. We will change a single permission type and see how it affects access without changing the keys.

1. Use Microsoft Azure Storage Explorer and drill down to container level of your storage account
2. Right-click on **container1** and select **Manage Access Policies**
3. On **SASPolicy1** uncheck **list** permission
4. Click **Save**
5. Check the same link as in previous task. Can you list the container objects now?

Note: By using SAS Tokens with Stored Access policies we got the possibility to change access to objects and containers without the need of changing the keys or generating a new SAS Token.

Activity 3: Using Storage Account File Service

Estimated time to complete this activity

15 minutes

Objectives

In this activity, you will learn how to create a new file share and how to access the stored data.

- ✦ You can use any existing general-purpose storage account, but it's recommended to use the one created in Activity 1
- ✦ We also strongly recommend that you use InPrivate browsing to ensure that you are not automatically logged on with other credentials during the registration / activation process.

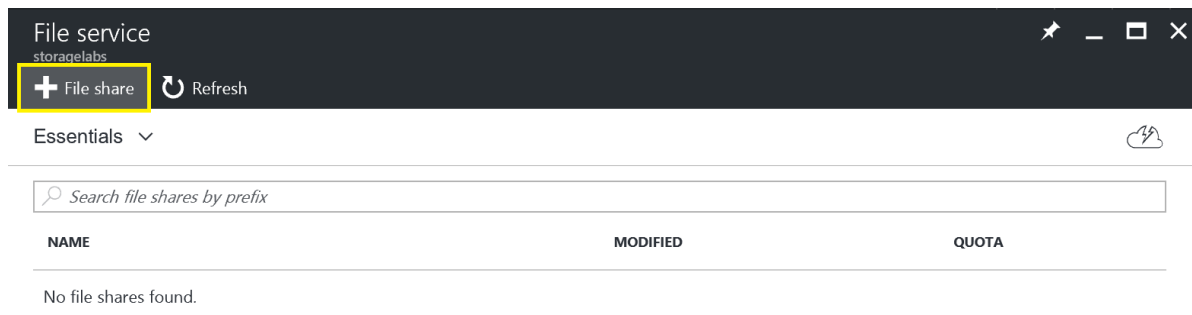
Exercise 1: Working with Azure Files

In this exercise, you will learn how to create a new file share and connect that file share to your client.

Create a new file share

In this task, you will learn how to create a new file share.

1. Open **Azure Portal**
2. Find the storage account created in Activity 1 / Exercise 2 and open Overview window (Storage Account should have been placed on main dashboard if 'Pin to dashboard' was selected, if not use search in Azure portal and type the name you defined in Activity 1 / Exercise 2).
3. From the list of services on the Overview window select '**Files**'
4. Add a file share by clicking '+ **File Share**'
Name: **fileshare** (name must contain small letters and numbers)
Quota: **5GB**
5. Click **Create**



New file share

File service (storagelabs)

* Name

fileshare

✓

Quota ⓘ

5

✓

GB

Create

Wait till the file share gets created.

Connect to file share

In this task, you will learn how to connect to a created file share from the client. Please note that for this access to work directly from the client, two requirements must be met: port 445 needs to be open to the internet and you must connect from the client that supports SMB3.0 and above. In case one of this requirement is not met, you can provision a VM in Azure and connect to the file share from that VM (in this case, you can use any OS that support SMB 2.1 and above).

1. Find the storage account created in Activity 1 / Exercise 2 and open Overview window (Storage Account should have been placed on main dashboard if 'Pin to dashboard' was selected, if not use search in Azure portal end type the name you defined in Activity 1 / Exercise 2).
2. From the list of services on the Overview window select '**Files**'
3. Click on the **fileshare**
4. Click on **Connect**
5. This will open a window with the instructions how to connect to the file share. Look at the instructions **Connecting from windows** as only Windows 8.1 and above support SMB3. If you will connect to the file share from VM running in Azure you can use both windows and linux OS.
6. On the local client, open run (windows + R) and type **cmd**
7. In command line type following command where <drive letter> is any drive letter available on the client, <storage account name> is storage account created in Activity 1 / Exercise 2 and <storage account access key> is either key1 or key2 we looked at Activity 1 / Exercise 2. For easy access to keys you can also take a look at the connect window of Azure Files in Azure Portal on the bottom of the page.

```
net use <drive letter> \\<storage account name>.file.core.windows.net\fileshare /u:<storage account name> <storage account access key>
```

(In our lab example this command would look like:

```
net use R: \\storagelabs.file.core.windows.net\fileshare /u:storagelabs
eXnSQIKHtu3G3Rj6lw56765FuWBhcuwS72V0cQuzDRkidKwl2qb5rEGOBWS2xtqm6xRtEEol0Fc
Wan11qmC5qw==
```

)
8. In case you see an error, most likely port 445 is not allowed or you are trying to connect from a client which doesn't support SMB3.
9. Open Windows Explorer and find the newly connected file share.
10. Copy any file to that file share
11. Use **Azure portal** to verify the file is uploaded by navigating to the file share.

Tip: To allow access to file share without the need to enter username and password every time you connect to the file share you can use the command where <storage account name> is storage account created in Activity 1 / Exercise 2 and <storage account access key> is either key1 or key2

```
Cmdkey /add:<storage account name>.file.core.windows.net /u:<storage account name>
/p:<storage account access key>
```

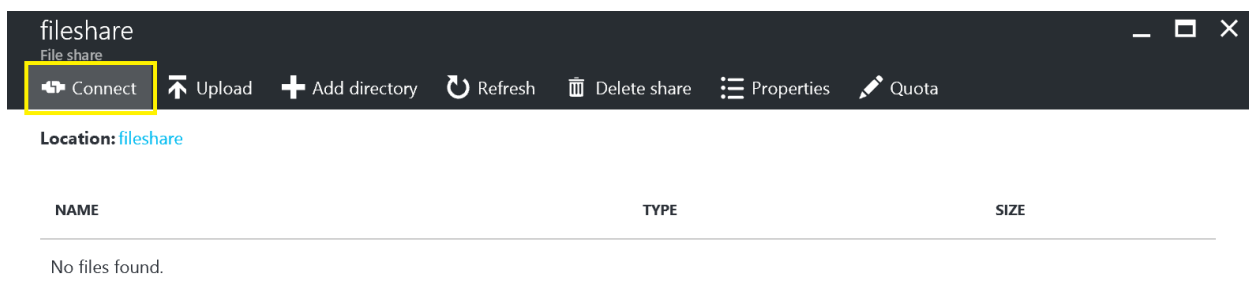
(in our lab example this command would look like:

```
cmdkey /add:storagelabs.file.core.windows.net /u:storagelabs
/p:eXnSQIKHtu3G3Rj6lw56765FuWBhcuwS72V0cQuzDRkidKwl2qb5rEgOBWS2xtqm6xRtEEol0FcWan
11qmC5qw==)
```

After using cmdkey you will connect to the share with command where <drive letter> is any drive letter available on the client, <storage account name> is storage account created in Activity 1 / Exercise 2

```
net use <drive letter> \\<storage account name>.file.core.windows.net\fileshare
```

(in our lab example this would be net use R: \\storagelabs.file.core.windows.net\fileshare)



Connecting from Windows

To connect to this file share from a Windows computer, run this command:

```
> net use [drive letter]  
\\storagelabs.file.core.windows.net\fileshare /u:storagelabs [storage  
account access key]
```

When connecting from a computer from outside Azure, remember to open outbound TCP port 445 in your local network. Some Internet service providers may block port 445. Check with your service provider for details.

[Learn more about Azure File Storage with Windows](#)

Connecting from Linux

To connect to this file share from a Linux computer, run this command:

```
> sudo mount -t cifs //storagelabs.file.core.windows.net/fileshare [mount  
point] -o vers=3.0,username=storagelabs,password=[storage account access  
key],dir_mode=0777,file_mode=0777
```

The Linux SMB3 client doesn't support share level encryption yet, so mounting a file share in Linux only works from virtual machines running in the same Azure region as the file share.

[Learn more about Azure File Storage with Linux](#)

[View access keys for this storage account](#)

Activity 4: Using StorSimple Virtual Array

Estimated time to complete this activity

120 minutes

Objectives

In this activity, you will learn how to create a StorSimple service manager and StorSimple virtual array, how to configure it and connect to the host.

- ✦ We also strongly recommend that you use InPrivate browsing to ensure that you are not automatically logged on with other credentials during the registration / activation process.

Exercise 1: Working with StorSimple

In this exercise, you will learn how to create a StorSimple service manager and create a StorSimple virtual array running in Azure. Please note that currently running StorSimple Virtual Array in Azure is not a supported configuration and this is used only for lab purposes.

Create a StorSimple service manager

In this task you will create a StorSimple service manager that is used to manage all StorSimple virtual devices.

1. Open **Azure Portal**
2. In Azure Portal search for **StorSimple** (Click on the '+' sign or press 'N' and type 'storsimple').
3. Select '**StorSimple Virtual Devices Series**'
4. Click '**Create**'
5. Enter the following information
 - a. Resource Name: StorSimple-Manager
 - b. Select appropriate subscription
 - c. Resource group: Create New and name it 'sslabs'
 - d. Location: West Europe
 - e. Leave the rest to default
6. Click '**Create**'
7. When finished go to created StorSimple Service manager and click on '+ **Virtual Array**'
8. Download the image named '**VHD for Hyper-V 2008 R2 and above**' or use a direct link <https://aka.ms/ss-vhd-2008>
9. Place it in any folder and unzip it

StorSimple Device Manager

Virtual device series

*

Resource name

StorSimple-Manager

✓

*

Subscription

Visual Studio Enterprise with MSDN

▼

*

Select a resource group. ⓘ

●

Create new

○

Use existing

sslabs

×

*

Location

West Europe

▼

☐

Create new Azure storage account

*

Storage account name

Enter name

☐

Pin to dashboard

Create

Automation options

StorSimple-Manager

StorSimple Device Manager

Search (Ctrl+/)

Virtual array

+ Add volume

+ Add share

Fail over

Delete

Overview

Activity log

Access control (IAM)

Tags

There are no registered devices. Click here to get started. →

Essentials ^

Resource group

sslabs

Location

West Europe

Subscription name

Visual Studio Enterprise with MSDN

Subscription ID

da01b8de-213e-4a65-9849-8d604a51c735

Provision a StorSimple Virtual Array

In this task you will provision a StorSimple Virtual Array in Azure.

1. Download provisioning scripts from the link <https://aka.ms/gbblabs-ss-scripts-upload>

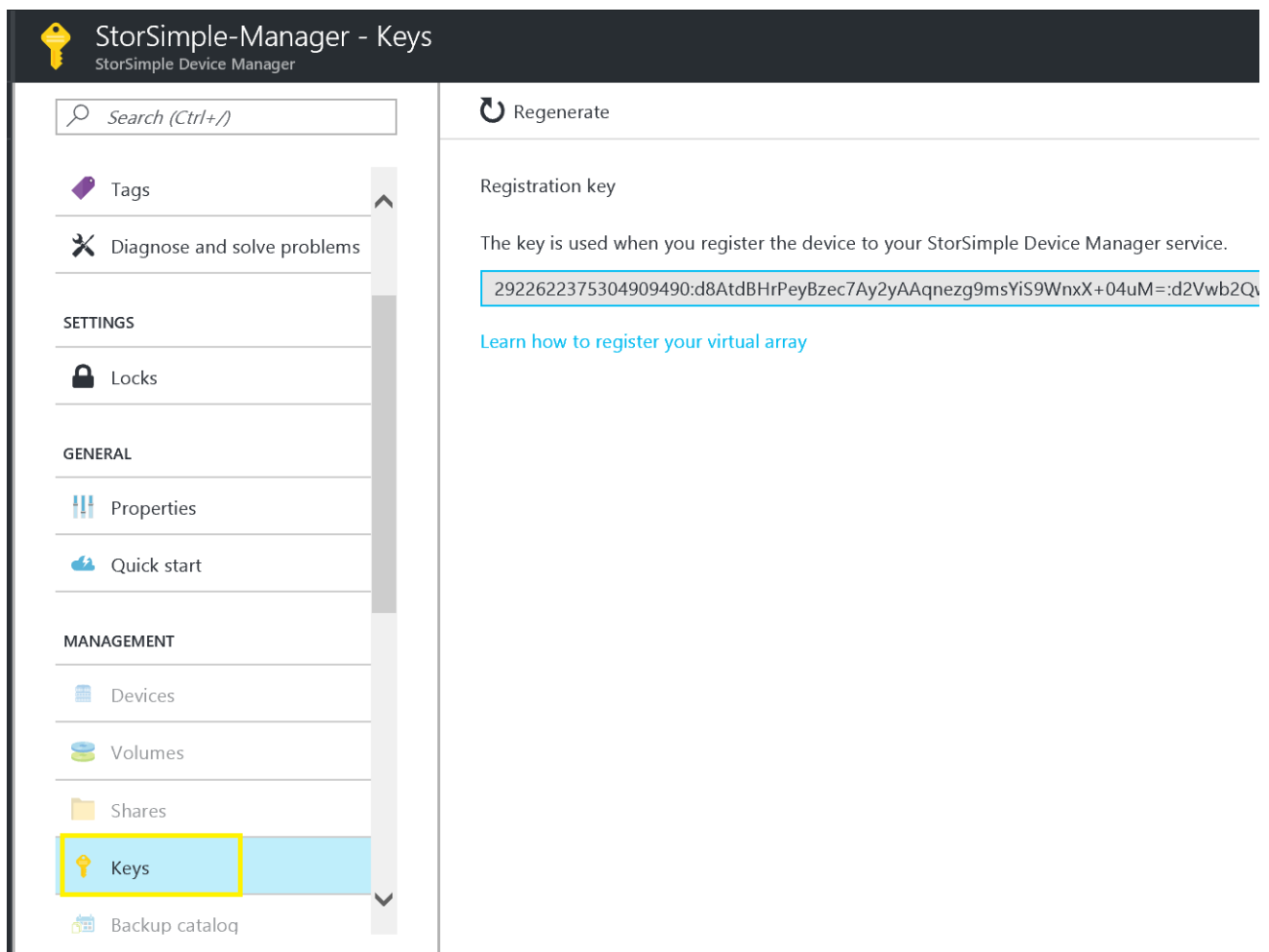
2. Unzip the file to the same folder where you placed the downloaded image from previous step
3. Open the file variables.ps1 with text editor
4. Change the following variables:
 - a. \$SSVName: Consult the proctor for the name as it needs to be unique
 - b. \$SubscriptionId: enter your subscription id. If the information is not known open Azure Portal, select **Resource Groups**, click on '**Columns**' and select to show Subscription ID
 - c. \$localVhdFile: make sure it matches the name of the downloaded and unzipped image
 - d. Check if IP address spaces for vNET, Subnet and private IP addresses if available. If not, change them accordingly.
5. Open Windows PowerShell ISE and change the directory to the one you unzipped the downloaded scripts
6. Run script 01-Login2Azure.ps1 to login to Azure
7. Run the script 02-PrepareImage.ps1 to create a storage account and upload the image to the storage account
8. Run script 03-CreateSSVA.ps1. This script will create a virtual machine from the copied image.

Register StorSimple Virtual Array

In this task you will register the newly provisioned virtual array.

1. Open **Azure Portal**
2. Search for a StorSimple Service Manager that was created in first step
3. From the menu on left side, select '**Keys**'
4. Copy Registration Key
5. Open a browser and connect to management link of StorSimple Virtual Array
https://<ssva_name>.westeurope.cloudapp.azure.com where ssva_name is name defined in a file variables.ps1 (\$SSVName). It takes couple of minutes after provisioning of the virtual array to access this link.
6. Login with the password Password1
7. Change password to a new password and remember it. Password must be a string that contains at least 8 characters, 3 or more of uppercase, lowercase, numeric, and special characters.
8. From the menu on the left check if '**Networking settings**' are ok. Virtual array should have two interfaces with IP addresses defined in variables.ps1 file (\$privIP01 and \$privIP02)

9. From the menu on the left select '**Device Settings**'. Select '**iSCSI server**' as device type, enter device name to match \$SSVName variable from the variables.ps1 file.
10. Click '**Apply**'
11. From the menu on the left select '**Time settings**' and change to your time zone
12. Click '**Apply**'
13. From the menu on the left select '**Cloud settings**' and enter registration key (from step 4)
14. Click '**Register**'.
15. A window will pop-up with Service Data Encryption Key. Copy it and paste to a text file as you will need it later to register the second device.
16. Wait till the virtual array reboots.



StorSimple-Manager - Keys
StorSimple Device Manager

Search (Ctrl+/)

Tags

Diagnose and solve problems

SETTINGS

Locks

GENERAL

Properties

Quick start

MANAGEMENT

Devices

Volumes

Shares

Keys

Backup catalog

Regenerate

Registration key

The key is used when you register the device to your StorSimple Device Manager service.

2922622375304909490:d8AtdBHrPeyBzec7Ay2yAAqnezg9msYiS9WnxX+04uM=:d2Vwb2Qv

[Learn how to register your virtual array](#)

Microsoft Azure

StorSimple 1200 DeviceSign out

Configuration

Get started

Network settings

Device settings

Web proxy settings

Time settings

Cloud settings

Maintenance

Power settings

Software update

Password change

Troubleshooting

Diagnostics tests

System logs

Contact Support

Enter the registration key to register this device to your StorSimple Manager service in Azure.

[More information on how to get the service registration key](#)

0cDovL3dpbmRvd3NjbG91ZGJhY2t1cC9tMw==:YWNjZXNzY29udHJvbmC53aW5kb3dzLm5ldA==:i7Jx50OxRt0gm1/t3LzPkA==:#adf8c1b469d14fb1✔

Enter the service data encryption key if this is not the first device. This key was generated when you successfully registered the first device with your service.

[More information on the service data encryption key](#)

Configure StorSimple Virtual Array

In this task, you will finish the configuration of the StorSimple virtual array using StorSimple service manager.

1. Open **Azure Portal**
2. Search for a StorSimple Service Manager that was created in previous task
3. From the menu on the left select '**Devices**'
4. Select the newly created device
5. Click '**Configure**'.
6. Enter Encryption key (32 characters).
7. Select the same storage account named the same as the virtual array or create a new one.
8. Click '**Add**'.
9. Once finished click on '**Configure**'.

Search (Ctrl+/)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

SETTINGS

Locks

GENERAL

Properties

Quick start

MANAGEMENT

Devices

Virtual array + Add volume + Add share

All statuses Total: 5

Filter items...

NAME	STATUS	REMAINING CAPACITY (LOCAL VS TIERED)	TYPE	MODEL
SSLABS01	Ready to set up	794.87 GB/7.76 TB	Virtual-iSCSI	1200

sslabs01

StorSimple-Manager

Settings

Configure

+ Add volume

Fail over

Take backup

More

Your device is ready to setup. Click the 'Configure' button above to complete setup.

Essentials

Status

Ready to set up

Device web UI

10.10.10.81

Model

1200

Device software version

10.0.10289.0

All settings →

Monitoring

Alerts - Past 7 days

0

There are no alerts.

Capacity

PROVISIONED

0 Bytes

REMAINING

7.76 TB

Tiered

794.87 GB

Local

Usage - Past 24 hours

Edit

Configure sslabs01	Storage account credentials	Add a storage account cred...
<div> Complete the configuration of an iSCSI server on this device to create volumes. </div> <p>Server name SSLABS01</p> <p>Cloud storage encryption ⓘ <input checked="" type="button" value="Enabled"/> <input type="button" value="Disabled"/> </p> <p>* Encryption key ⓘ <input type="text" value="....."/> ✓ </p> <p>* Confirm encryption key <input type="text" value="....."/> ✓ </p> <div> <p>* Storage account credential Configure required settings</p> </div> <p><input type="button" value="Configure"/></p>	<div> There are no storage account credentials. Add a new storage account credential. </div> <div> Add new </div> <p>No items found.</p>	<p>Subscription ⓘ <input checked="" type="button" value="Current"/> <input type="button" value="Other"/> </p> <div> <p>* Storage account ⓘ sslabs01 </p> <p>* Location ⓘ West Europe </p> <p>Enable SSL ⓘ <input checked="" type="button" value="ENABLE"/> <input type="button" value="DISABLE"/> </p> <p><input type="button" value="ADD"/></p> </div>

Provision a Windows host

In this task, you will provision a new windows host.

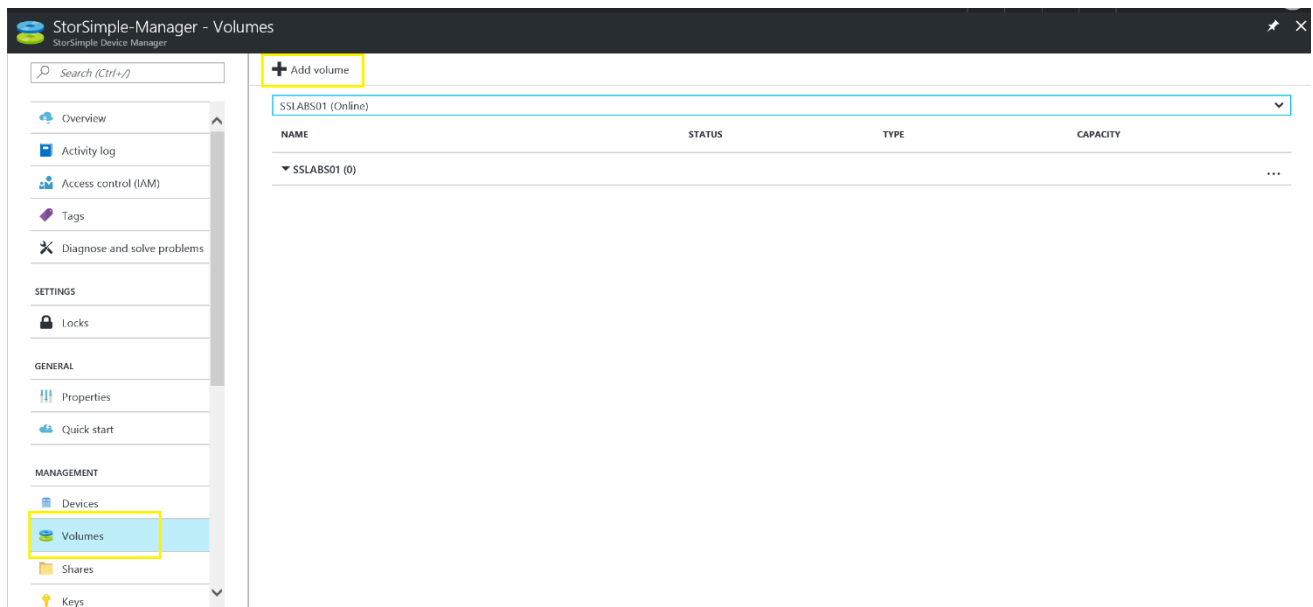
1. Open **Azure Portal**
2. Provision a windows virtual machine
3. Select same vNET and Subnet as StorSimple virtual array
4. Connect to a virtual machine
5. Select **Server Manager**
6. Go to **Manage – Add Roles and Features**
7. Install **Multipath I / O** feature
8. From the **Server Manager – Tools** select **iSCSI Initiator** and start the service
9. Go to **Configuration** tab and copy **Initiator name**
10. Open **MPIO** from **Server Manager – Tools**
11. Go to **Discover Multi-Paths** tab
12. Check **Add support for iSCSI devices** and click **Add**

13. Reboot the server

Add a new volume

In this task you will create a new volume on newly created StorSimple Virtual Array.

1. Open **Azure Portal**
2. Search for a StorSimple Service Manager that was created in previous task
3. Select '**Volumes**'.
4. Click on '+ **Add volume**'.
 - a. Select device created in previous step
 - b. Enter volume name: vol01
 - c. Capacity: 500GB
5. Click on '**Connected hosts**' and select '**Add new**'
 - a. Name: windows
 - b. IQN: copy the iqn from previous task
6. Click '**Add**'
7. Select '**Windows**' and click '**Select**'
8. Click '**Create**'



The screenshot shows three windows from the StorSimple Manager interface:

- Add volume:**
 - Storage pool: SSLABS01
 - Volume name: vol01
 - Type: Tiered
 - Capacity: 500 GB
 - Buttons: Create, Connected hosts (Select)
 - Info: Available total capacity : 7.76 TB. Available local capacity : 794.87 GB. For a tiered volume, 10% of the provisioned volume size or 200 GB, whichever is lower is reserved locally on the device.
 - Backup info: Backup is automatically enabled on this volume. You can modify the start time of the backups from the Backup policy blade under Device settings.
- Connected hosts:**
 - Buttons: Add new
 - Status: No items found.
 - Button: Select
- Add ACR:**
 - Name: windows
 - IQN: iqn.1991-05.com.microsoft:sslabs-win01
 - Buttons: ADD

Mount a volume to the windows host

In this task, you will connect the created volume to a windows host

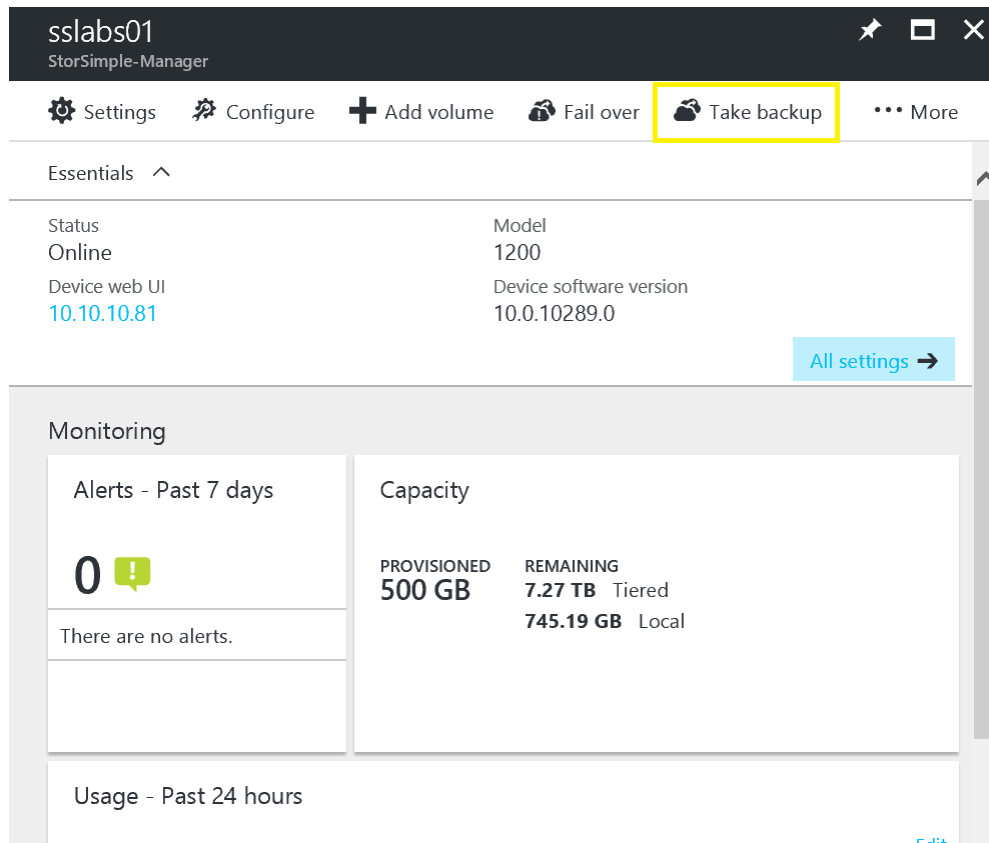
1. Connect to a virtual machine
2. Select **Server Manager**
3. From **Tools** select **iSCSI Initiator**
4. Go to **Discovery** tab and click on **Discover Portal**
5. Enter the IP address of StorSimple virtual array (either \$privIP01 or \$privIP02 from variables.ps1 file) and click **OK**
6. Go to **Targets** tab
7. Select the windows IQN and click **Connect**
8. Check **Enable multi-path**
9. Select **Advanced**
 - a. Local Adapter: Microsoft iSCSI Initiator
 - b. Initiator IP: IP address of the windows host
 - c. Target portal IP: Select the first IP address of virtual array (\$privIP01 from variables.ps1)
10. Click **OK** on both windows
11. Select windows IQN and click **Properties**
12. Click **Add session**

13. Check **Enable multi-path**
14. Select **Advanced**
 - a. Local Adapter: Microsoft iSCSI Initiator
 - b. Initiator IP: IP address of the windows host
 - c. Target portal IP: Select the second IP address of virtual array (\$privIP02 from variables.ps1)
15. Click **OK** on all windows
16. Open **Disk management** (right mouse click on windows sign)
17. You should see the 500 GB drive from the virtual array
18. Bring it online and format the drive by creating New Simple Volume (change Allocation unit size to 64KB)
19. Open the drive and copy some data on it

Create a cloud backup

In this task you will create a new cloud snapshot on the StorSimple Virtual Array.

1. Open **Azure Portal**
2. Search for a StorSimple Service Manager that was created in previous task
3. From the menu on the left, select '**Devices**' and click on the virtual array name
4. Click on '**Take Backup**'.
5. Monitor the job from the StorSimple Service Manager (Jobs menu on the left side)
6. When the job is done go back to the windows VM and delete the data we copied on the StorSimple volume



Restore the volume

In this task, you will restore the data deleted in the previous step from the cloud snapshot

1. Open **Azure Portal**
2. Search for a StorSimple Service Manager that was created in previous task
3. Select '**Backup catalog**' from the menu on the left
4. Expand your virtual array and select '**vol01**'
5. Click on the '**...**' on the right side and select '**Clone**'.
6. Select the same host from the '**Connected hosts**' option
7. Click '**Clone**'
8. Monitor the job from the Jobs menu on the left side
9. Once finished, connect to the windows virtual machine
10. Open **Disk Management** and select **Rescan Disks** from **Action** menu. If you don't see a new volume configure new iSCSI connection (repeat steps 6 – 17 from the "Mount a volume to the windows host" task).
11. Open the new volume and confirm that the deleted data is there.

Failover to a new device

In this task, you will learn how to do a failover from one virtual array to another

1. Provision a new virtual array
 - a. Change the values for \$SSVName, \$privIP01, \$privIP02 from the file variables.ps1
 - b. Follow the steps 3 – 8 from the task "Provision a StorSimple Virtual Array"
 - c. Follow the steps 1 – 13 from the task "Register StorSimple Virtual Array" but before clicking on '**Register**' you need to enter Service Data Encryption Key as well (key copied from the registering first device)
2. Connect to a windows virtual machine and disconnect all the connected drives from the first virtual arrays (from the iSCSI Initiator select Targets tab, select connections one-by-one and click on **Disconnect**. Use Disk Management to confirm volumes are no longer attached to a host. Remove the IP address of the first device from the **Discovery** tab).
3. Open **Azure Portal**
4. Search for a StorSimple Service Manager that was created in previous task
5. From the menu on the left, select '**Devices**'
6. Select the first virtual array name
7. Click on '...' on the right side and select '**Deactivate**'.
8. Enter name of the device and click '**Deactivate**'.
9. Click on '...' on the right side and select '**Failover**'. StorSimple will take the latest cloud snapshot and do a failover to a newly created device.
10. Monitor the failover job until it is finished. It should take couple of minutes.
11. When finished, confirm that the first device is deleted and that the second device contains the volume vol01 and it has the same storage account attached as the first device
12. Connect to the windows virtual machine and connect to the new device (repeat steps 1 – 18 from the task "Mount a volume to the windows host" but use the IP addresses from the second device).
13. Open the volume to confirm that the data we copied on the StorSimple volume is there.