



**ProfileUnity™  
with FlexApp™ Technology**

***Using Cloud Storage  
with ProfileUnity***

## Introduction

This guide has been authored by experts at Liquidware in order to provide information and guidance concerning the use of cloud storage with ProfileUnity with FlexApp.

Information in this document is subject to change without notice. No part of this publication may be reproduced in whole or in part, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any external use by any person or entity without the express prior written consent of Liquidware Labs.

### **Liquidware Labs, Inc.**

3600 Mansell Road

Suite 200

Alpharetta, Georgia 30022

U.S.A.

Phone: 678-397-0450

[www.liquidware.com](http://www.liquidware.com)

*©2019 Liquidware Labs Inc. All rights reserved. Stratusphere, ProfileUnity, FlexApp, FlexDisk and ProfileDisk are trademarks of Liquidware Labs. All other products are trademarks of their respective owners. 19-0416.6.8.1*

# Contents

- OVERVIEW ..... 4**
- SUPPORTED CLOUD STORAGE OBJECTS ..... 4**
- AMAZON S3 ..... 5**
  - SETTING UP AN AMAZON S3 BUCKET ..... 5
  - CREATING FOLDERS ..... 8
  - GRANTING ACCESS TO THE PROFILEUNITY MANAGEMENT CONSOLE AND CLIENT ..... 8
  - PUTTING THIS ALL TOGETHER ..... 12
- AZURE BLOB ..... 13**
  - CREATING A CONSOLE CONFIGURATION STORAGE ACCOUNT ..... 13
  - CREATING A CONFIGURATION CONTAINER ..... 14
  - GETTING THE ACCESS KEY FOR CONFIGURATION ..... 14
  - CREATING A CLIENT PORTABILITY STORAGE ACCOUNT ..... 15
  - CREATING A PORTABILITY CONTAINER ..... 16
  - GETTING THE ACCESS KEY FOR PORTABILITY ..... 16
  - RESTRICTING PROFILEUNITY CLIENT ACCESS TO READ-ONLY FOR CONFIGS ..... 17
  - PUTTING THIS ALL TOGETHER ..... 18
- GOOGLE CLOUD STORAGE ..... 19**
  - CREATING A CONSOLE SERVICE ACCOUNT ..... 19
  - CREATING A CLIENT SERVICE ACCOUNT ..... 20
  - CREATING A CONFIGURATION BUCKET ..... 20
  - CREATING A PORTABILITY BUCKET ..... 21
  - SETTING PERMISSIONS ON THE PORTABILITY BUCKET ..... 23
  - PUTTING THIS ALL TOGETHER ..... 23
- GETTING HELP INSTALLING PROFILEUNITY ..... 25**
  - USING ONLINE RESOURCES ..... 25
  - CONTACTING SUPPORT ..... 25



## Overview

In keeping with Liquidware's mission to provide flexible, scalable, and complete Workspace Environment Management (WEM), ProfileUnity provides the option to connect its user profile solution with cloud storage. Cloud storage offers organizations many advantages including disaster recovery, continuity of operations, and greater security in accordance with industry regulations. ProfileUnity utilizes cloud storage for use with configuration files and user profiles. The purpose of this document is to give guidance on how to setup cloud storage to work with your ProfileUnity solution.

## Supported Cloud Storage Objects

The following table shows which cloud storage providers are supported by ProfileUnity and which features or components are able to utilize cloud storage.

ProfileUnity Feature/Component	Amazon Simple Storage Service (S3)	Microsoft Azure Blob Storage	Google Cloud Storage
Management Console Configuration Files	Yes	Yes	Yes
License File	Yes	Yes	Yes
User Portability Files	Yes	Yes	Yes
FlexApp	Yes	Yes	Yes
FlexDisk	No	No	No
ProfileDisk	Coming soon	Coming soon	Coming soon

## Amazon S3

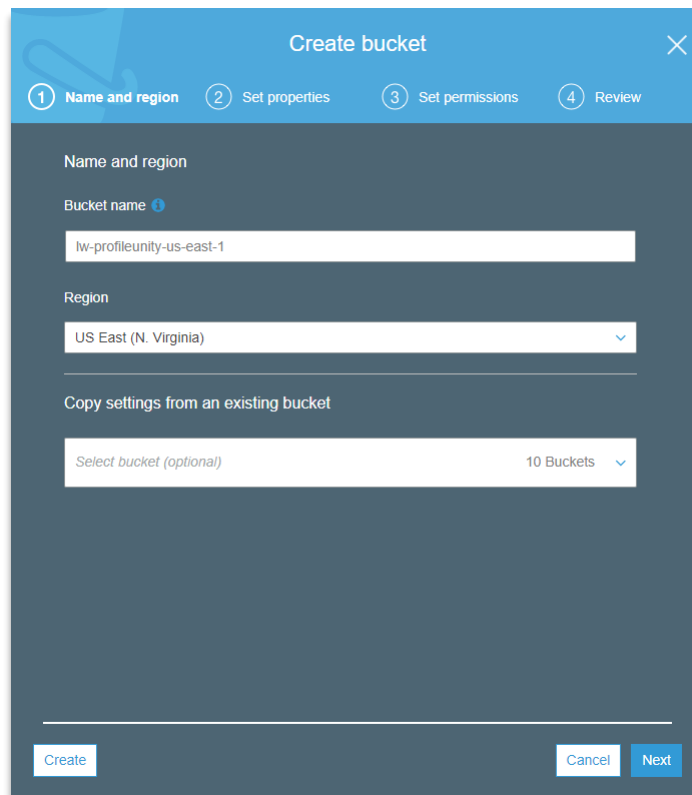
Amazon S3 provides object storage to access data over the internet. The data is stored inside a resource called a “bucket”. Each bucket can hold as many objects as you want. Administrators control access to their storage bucket and who can read, write, or delete data objects in that bucket. For more detailed information, please visit the [Amazon Web Services](#) website.

### Setting up an Amazon S3 Bucket

After setting up an AWS Account, you will need to create an Amazon S3 bucket to store ProfileUnity files using the following steps:

1. Login to the AWS Management Console and open the Amazon S3 console.
2. Click on **Create Bucket**.
3. For Step 1, name your storage bucket and select the AWS Region for your bucket location.

Complete these fields using the following guidance:



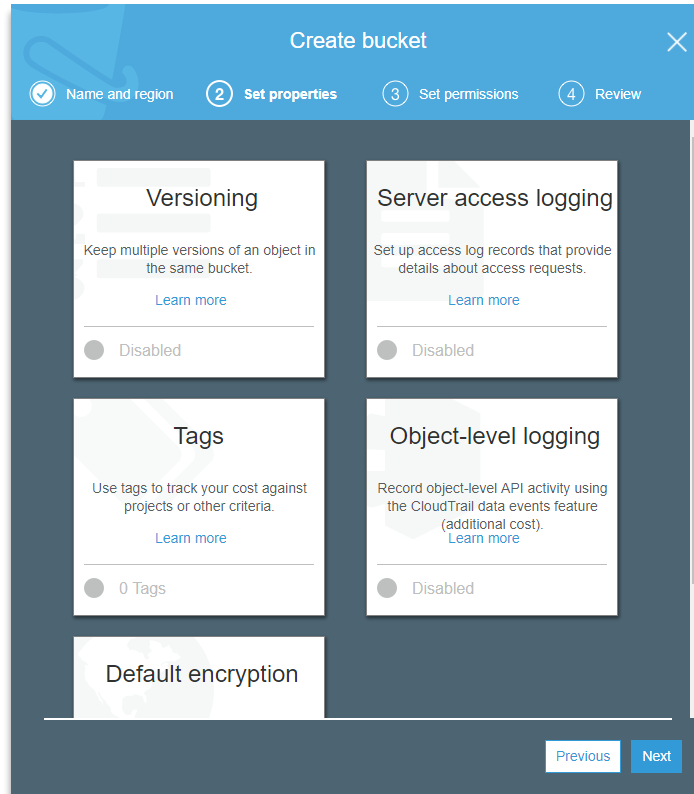
The screenshot shows the 'Create bucket' wizard in the AWS Management Console. The title bar is blue with a close button (X) on the right. Below the title bar is a progress indicator with four steps: 1. Name and region (active), 2. Set properties, 3. Set permissions, and 4. Review. The main content area is dark blue and contains the following fields:

- Name and region**
  - Bucket name**: A text input field containing 'lw-profileunity-us-east-1'.
  - Region**: A dropdown menu showing 'US East (N. Virginia)'.
- Copy settings from an existing bucket**: A dropdown menu showing 'Select bucket (optional)' and '10 Buckets'.

At the bottom of the form, there are three buttons: 'Create' (white with blue border), 'Cancel' (white with blue border), and 'Next' (blue with white text).

- a. **Type in your Bucket name.** All bucket names must be unique across all regions. Therefore, the recommended name format is `yourcompanyname-bucketuse-bucketregion` where “bucketuse” is what this particular bucket is being used for (in this case, ProfileUnity) and “bucketregion” is the AWS region where this bucket will be stored. Here are a few additional naming requirements:
  - i. The name must be unique across all existing bucket names in Amazon S3.
  - ii. The name must not contain uppercase characters.
  - iii. The name must start with a lowercase letter or number.
  - iv. The name must be between 3 and 63 characters long.

- v. After you create the bucket you cannot change the name, so choose wisely.
  - vi. Choose a bucket name that reflects the objects in the bucket because the bucket name is visible in the URL that points to the objects that you're going to put in your bucket.
- b. **Select the Region where the bucket will be located.** Typically, customers choose a region location that is close in proximity to reduce latency and costs, or to meet regulatory requirements. Refer to the AWS website for a list of [Amazon S3 Regions and Endpoints](#).
- c. Click **Next**.
4. For Step 2 – Set properties, no changes in settings are needed. Click **Next**.



5. For Step 3 – Set permissions, configure the following:

The screenshot shows the 'Create bucket' wizard in the AWS console, specifically the 'Set permissions' step. The progress bar at the top indicates four steps: 1. Name and region, 2. Set properties, 3. Set permissions (current), and 4. Review. The 'Manage users' section shows a table with columns for 'User ID', 'Objects', and 'Object permissions'. The user 'jon.mcdonald(Owner)' is listed with 'Read' and 'Write' permissions checked for both 'Objects' and 'Object permissions'. There is an 'Add account' button for 'Access for other AWS account'. The 'Manage public permissions' section has a dropdown menu set to 'Do not grant public read access to this bucket (Recommended)'. The 'Manage system permissions' section has a dropdown menu set to 'Do not grant Amazon S3 Log Delivery group write access to this bucket'. At the bottom right, there are 'Previous' and 'Next' buttons.

- Leave the default owner with full read and write access.
- Set **Manage public permissions** to “Do not grant public read access to this bucket”.
- Set **Manage system permissions** to “Do not grant Amazon S3 Log Delivery group write access to this bucket”.
- Click **Next**.



6. For Step 4 - Review, check your settings. They should look similar to this example. Click **Create bucket** when you are ready to commit your configuration settings.

The screenshot shows the 'Create bucket' wizard in the AWS S3 console, currently on the 'Review' step. The progress bar at the top indicates that 'Name and region', 'Set properties', and 'Set permissions' are completed, while 'Review' is the current step. The configuration details are as follows:

Name and region		Edit
Bucket name	lw-profileunity-us-east-1	Region US East (N. Virginia)

Properties		Edit
Versioning	Disabled	
Server access logging	Disabled	
Tagging	0 Tags	
Object-level logging	Disabled	
Default encryption	None	

Permissions		Edit
Users	1	
Public permissions	Disabled	
System permissions	Disabled	

At the bottom of the form, there are two buttons: 'Previous' and 'Create bucket'.

## Creating Folders

After the ProfileUnity bucket is setup, you will want to create folders for different types of data files.

The Amazon S3 structure provides a very flat file structure. Each bucket created simply holds your data objects. You cannot have a hierarchy of buckets inside buckets or subbuckets. However, you can use the Amazon S3 Console to emulate a view of subfolders inside your ProfileUnity bucket.

From the Amazon S3 Console, select the ProfileUnity bucket you created. Then click on the **Create Folder** button and enter the name of your folder. Repeat until all of the following folders have been created:

- configurations
- portability
- flexapp

## Granting Access to the ProfileUnity Management Console and Client

In order for ProfileUnity to make use of the Amazon S3 cloud storage, ProfileUnity will have to login to AWS. We do not want to use the default or root AWS admin account. Instead, we need to setup two separate accounts for ProfileUnity – one for the management console and one for the client.

AWS Identity and Access Management (IAM) allows you to securely manage access to all AWS services. IAM allows you to create AWS users and groups while using permissions and roles to control user account access to data. AWS does not charge per IAM account. It only charges for the use of AWS services by those user accounts. Creating two separate AWS users for ProfileUnity allows you to fine-tune permissions and audit ProfileUnity Management Console and Client activity more closely.

To create the ProfileUnity Management Console’s AWS user account:

1. Login to the AWS Management Console and open the IAM console.
2. From the left-hand navigation, select **Users**. Then click on the **Add User** button at the top of the user list.
3. Set the user details:

**Add user** 1 2 3 4

**Set user details**

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\* profileunity-console

+ Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type\*  **Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

**AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

\* Required

Cancel **Next: Permissions**

- a. Create a user name for the management console user such as “profileunity-console”.
  - b. Check **Access type: Programmatic access** for this user account.
  - c. Click **Next: Permissions**.
4. To set permissions for this user, click **Attach existing policies directly**, and complete the following:

**Add user** 1 2 3 4

**Set permissions for profileunity-console**

Add user to group Copy permissions from existing user **Attach existing policies directly**

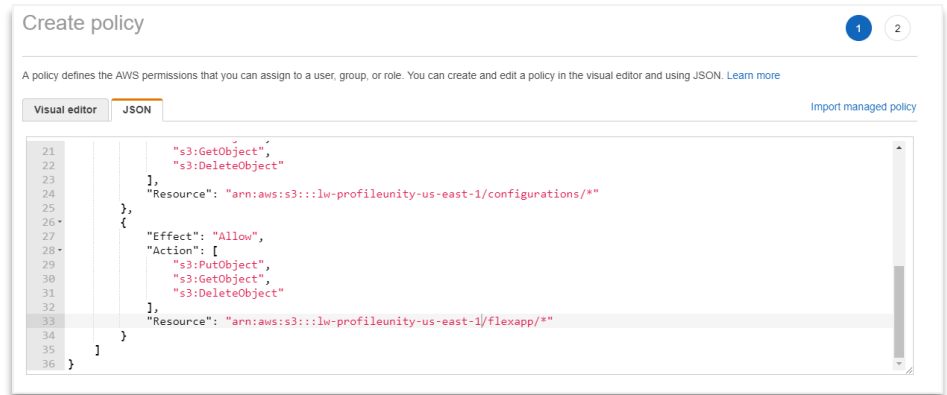
Attach one or more existing policies directly to the users or create a new policy. [Learn more](#)

Create policy Refresh

Filter: Policy type Search Showing 335 results

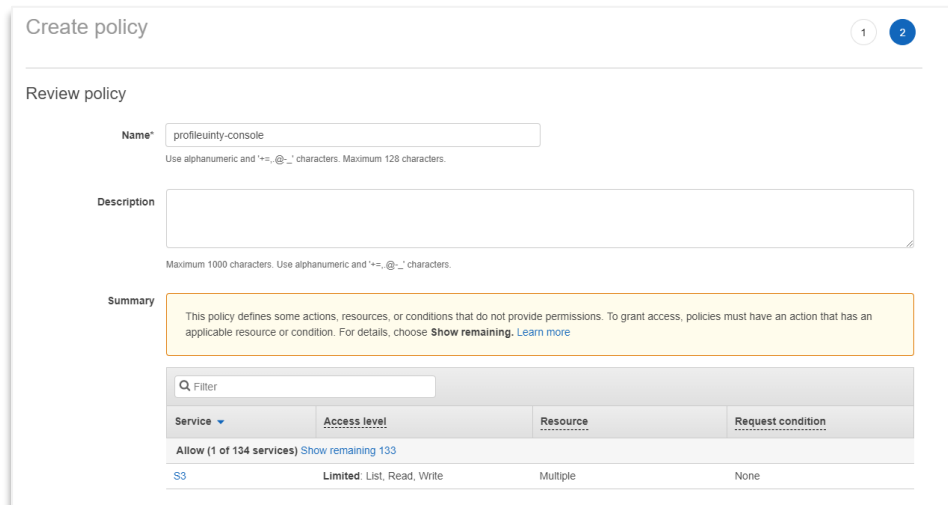
Policy name	Type	Attachments	Description
-------------	------	-------------	-------------

- a. Click **Create Policy**.



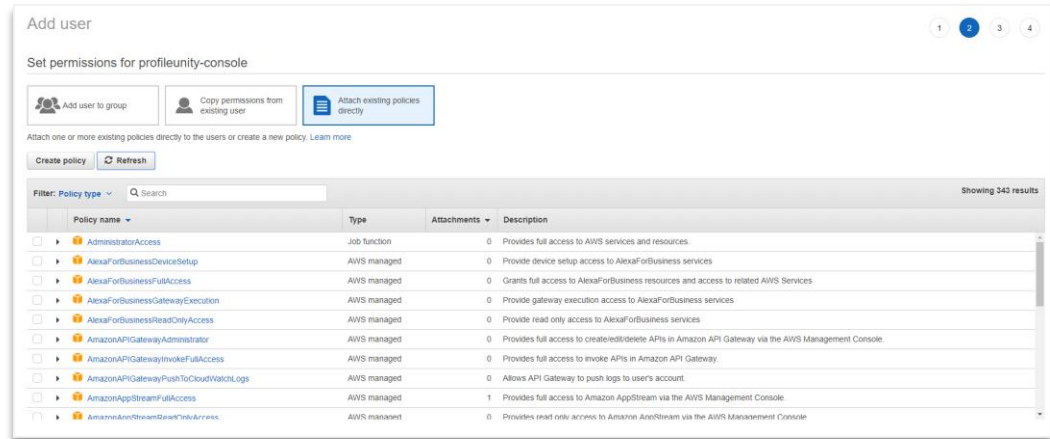
- b. In the new tab that opens, click on the **JSON** tab.
  - c. Copy the following JSON code:
 

<http://download.liquidwarelabs.com/ProfileUnity.NET/AWS-ProfileUnityConsole-UserPolicyv1.1.txt>
  - d. Paste the JSON code into the JSON policy tab. Once the code is in the policy tab, you will need to edit the Amazon S3 bucket information on lines 15, 24, and 33 to match the ProfileUnity bucket name you created earlier.
  - e. Click **Review policy**.
5. Give the policy a name such as “profileunity-console”, and click **Create Policy**.

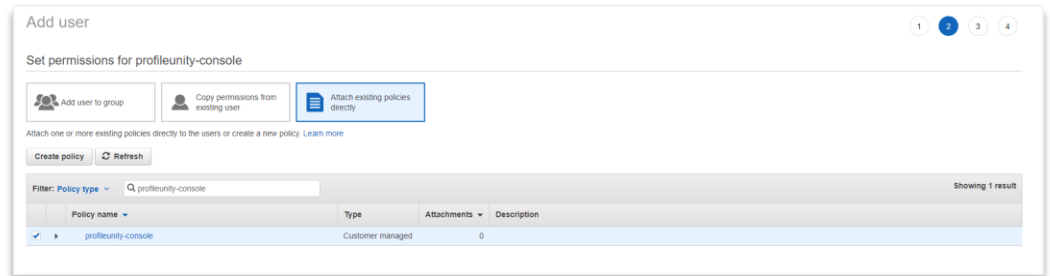


6. After clicking Create Policy, you will need to go back to the original browser tab to finish creating the user.

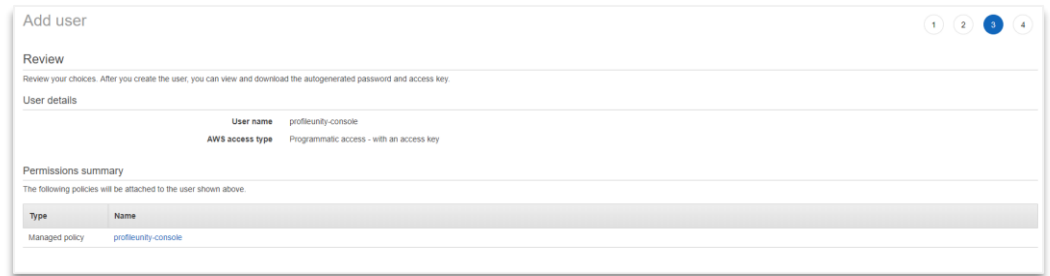
- Go back to the Set Permissions step for the user and click the **Refresh** button to update the policy list.



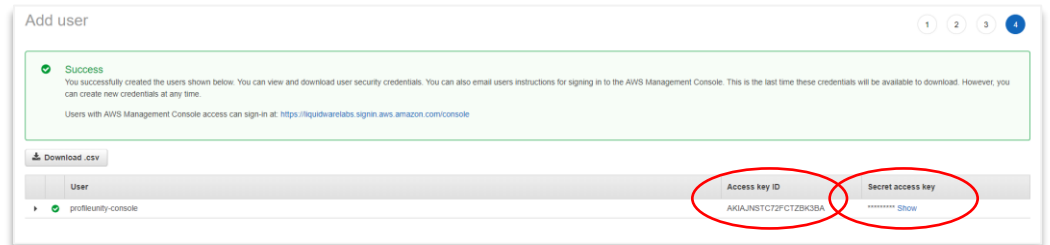
- In the Search field, type in the name of the policy you just created (i.e. profileunity-console). When you find the new policy in the list, check the box next to its name and click **Next:Review**.



- Review the information for the new user and click **Create User**.



- The new user will be created. At this time, you will be given the user's **Access Key ID** and **Secret Access Key**. Make sure to document these credentials and keep them in a safe place. You may also download them as a CSV. If you forget or lose these keys, you will NOT be able to access them again. However, you can create a new Access and Secret key to reset your credentials.



To create the ProfileUnity Client's AWS user account:

1. Follow the same steps that you used above for creating the Management Console's AWS user account.
2. Use a descriptive name for the user account such as "profileunity-client".
3. You will need to copy different policy code for the Client user account. Copy and paste the following code into the JSON policy tab:
  - a. <http://download.liquidwarelabs.com/ProfileUnity.NET/AWS-ProfileUnityClient-UserPolicyv1.1.txt>
4. Once the code is in the policy tab, you will need to edit the Amazon S3 bucket information on lines 15, 20, 25, and 34 to match the ProfileUnity bucket name you created earlier.
5. Remember to search for and select the new "profileunity-client" policy for this new ProfileUnity Client user.
6. Once the client user has been created, be sure to document and save the Access Key ID and the Secure Access Key.

## Putting This All Together

After completing all of these instructions to setup your Amazon S3 cloud storage for ProfileUnity, you will be able to install and configure ProfileUnity to make use of your new cloud storage bucket.

When using a cloud storage template, ProfileUnity's Guided Configuration Wizard will prompt you for your Amazon S3 bucket name, your ProfileUnity Console IAM user account credentials, and your ProfileUnity Client IAM user account credentials.

Please note that when ProfileUnity refers to Amazon S3 cloud storage paths, they begin with "S3://". Here are some examples where what is in brackets is replaced with the specified bucket name:

- **Deployment/Console Path:** S3://{bucket}/configurations
- **Portability/Client Path:** S3://{bucket}/portability
- **FlexApp Packages Path:** S3://{bucket}/flexapp
- **GPO Settings for INI, ProfileUnity as a Service, Client Settings XML Path:**  
S3://{bucket}/configurations

You may change your ProfileUnity Console or Client IAM credentials at any time by going to the Cloud Storage Settings section in the ProfileUnity Management Console Administration Area.

Please see our ProfileUnity Installation and Configuration Guide for more instructions on how to adjust your Licensing and GPO configuration to utilize cloud storage.

## Azure Blob

Azure Blob is Microsoft’s object storage solution for accessing data over the internet. The data is stored inside a resource called a “blob”. Each blob can hold as many objects as you want. Blobs are organized inside of “containers” within a storage account. Each container can hold an unlimited number of blobs. Administrators control access to their storage blobs and who can read, write, or delete data objects in that blob. For more detailed information, please visit the [Microsoft Azure Blob Storage](#) website.

### Creating a Console Configuration Storage Account

The Configuration storage account holds the configuration files created in the ProfileUnity Management Console and the license file. The ProfileUnity Client reads these files to configure a user’s desktop.

1. Login to your Azure portal.
2. In the left-hand navigation, go to **More Services** and select **Storage > Storage Accounts**. Alternatively, you can type “Storage Accounts” in the Search bar.
3. Click on **+Add** to add the new storage account.
4. Configure the storage account using the following guidance:

The screenshot shows the 'Create storage account' form in the Azure portal. The form is titled 'Create storage account' and includes the following fields and options:

- Name:** lvprouconfigs (with a checkmark icon)
- Deployment model:** Resource manager (selected), Classic
- Account kind:** Blob storage (selected)
- Location:** East US (selected)
- Replication:** Read-access geo-redundant storage (R... (selected)
- Performance:** Standard (selected), Premium
- Access tier (default):** Cool (selected), Hot
- Secure transfer required:** Disabled, Enabled (selected)
- Subscription:** Visual Studio Enterprise (selected)
- Resource group:** Create new (selected), Use existing; lvprouconfigs (with a checkmark icon)
- Virtual networks:** Configure virtual networks; Disabled, Enabled (selected)
- Pin to dashboard:**
- Create:** Button
- Automation options:** Link

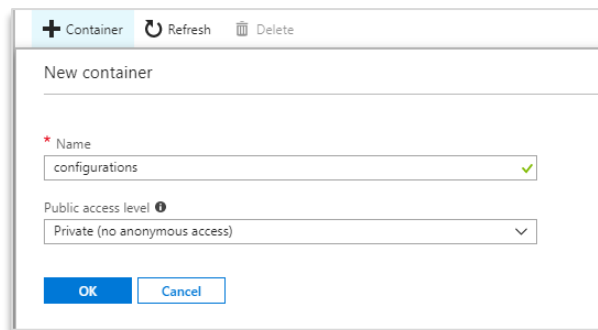
- a. Name this storage account. The name you select must be unique across all Azure storage accounts. Therefore, best practice is to use a format such as

`yourcompanynamepurpose`, for example, “lvprouconfigs”. Here are a few additional naming requirements:

- i. The name must be between 3 and 24 characters long.
  - ii. The name can only use lowercase letters and numbers.
5. Select “Blob storage” for the **Account kind**.
  6. Select your **Location**. This should be the same location as your desktops ProfileUnity will manage.
  7. Please read more about your **Replication options**, and then make a selection.
  8. Select “Hot” for your **Access tier** because this is data that will be referenced often.
  9. Either create a new **Resource Group** or use an existing one. For this example, we created a new one.
  10. The configuration of virtual networks is not required.
  11. Click **Create** when done.

## Creating a Configuration Container

1. After the ProfileUnity Configuration Storage Account is created, the Azure portal will display information about the new storage account. Click on the **Blobs** tile.
2. Click on **+Container** to add a container to the storage account.
3. Name the container “configurations”, and set the **Public access level** to “Private”. Then click **OK**.

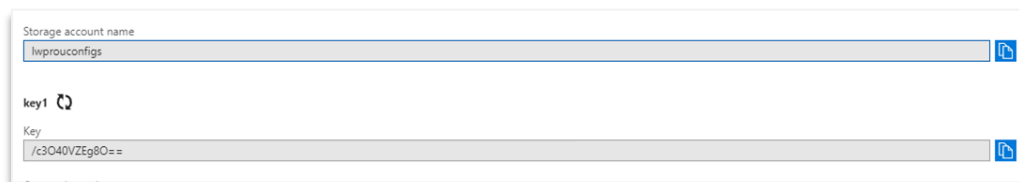


The screenshot shows a dialog box titled "New container" with a header bar containing "+ Container", "Refresh", and "Delete" icons. Below the header, there is a text input field for "Name" with the value "configurations" and a green checkmark icon to its right. Below that is a dropdown menu for "Public access level" with the selected option "Private (no anonymous access)". At the bottom of the dialog are two buttons: "OK" and "Cancel".

## Getting the Access Key for Configuration

The storage access key is used to authenticate access to the storage account. This permits full access to all containers and blobs in this account.

1. In the Azure portal, navigate back to the ProfileUnity Configuration Storage Account.
2. Under **Settings**, click on **Access Keys**.
3. Copy the **Storage account name** and the **Key** and save this information until you are ready to enter it into the ProfileUnity Management Console. In the ProfileUnity Management Console, the Account Name and Account Key credentials for Microsoft Azure can be entered as the Console Credentials either in the Cloud Storage section of Administration Settings or in the Guided Configuration Wizard when using a cloud storage template.

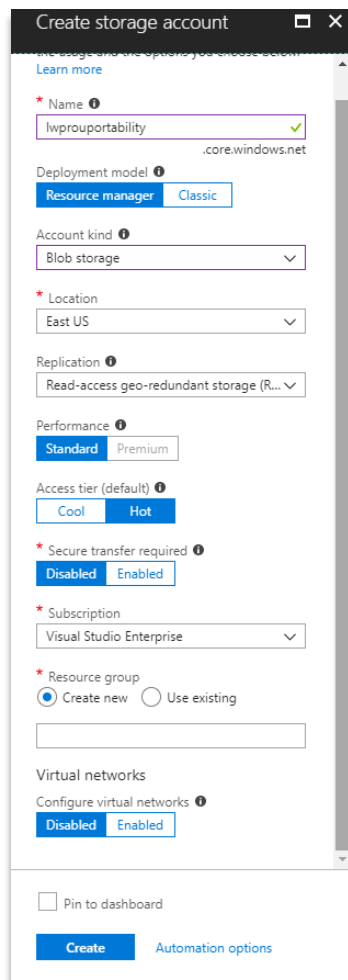


The screenshot shows a form with two input fields. The first field is labeled "Storage account name" and contains the text "lvprouconfigs". The second field is labeled "key1" and contains the text "/c3Q40VZEg8O==". Both fields have a copy icon to their right.

## Creating a Client Portability Storage Account

The Portability storage account holds a user's profile settings that the ProfileUnity Client writes back to it.

1. Login to your Azure portal.
2. In the left-hand navigation, go to **More Services** and select **Storage > Storage Accounts**.  
Alternatively, you can type "Storage Accounts" in the Search bar.
3. Click on **+Add** to add the new storage account.
4. Configure the storage account using the following guidance:



- a. Name this storage account. The name you select must be unique across all Azure storage accounts. Therefore, best practice is to use a format such as *yourcompanynamepurpose*, for example, "lwproutability". Here are a few additional naming requirements:
    - i. The name must be between 3 and 24 characters long.
    - ii. The name can only use lowercase letters and numbers.
5. Select "Blob storage" for the **Account kind**.
  6. Select your **Location**. This should be the same location as your desktops ProfileUnity will manage.
  7. Please read more about your **Replication options**, and then make a selection.
  8. Select "Hot" for your **Access tier** because this is data that will be referenced often.



9. Either create a new **Resource Group** or use an existing one. For this example, we created a new one.
10. The configuration of virtual networks is not required.
11. Click **Create** when done.

## Creating a Portability Container

1. After the ProfileUnity Configuration Storage Account is created, the Azure portal will display information about the new storage account. Click on the **Blobs** tile.
2. Click on **+Container** to add a container to the storage account.
3. Name the container “configurations”, and set the **Public access level** to “Private”. Then click **OK**.

## Getting the Access Key for Portability

The storage access key is used to authenticate access to the storage account. This permits full access to all containers and blobs in this account.

1. In the Azure portal, navigate back to the ProfileUnity Configuration Storage Account.
2. Under **Settings**, click on **Access Keys**.
3. Copy the **Storage account name** and the **Key** and save this information until you are ready to enter it into the ProfileUnity Management Console. In the ProfileUnity Management Console, the Account Name and Account Key credentials for Microsoft Azure can be entered as the Client Credentials either in the Cloud Storage section of Administration Settings or in the Guided Configuration Wizard when using a cloud storage template.

## Restricting ProfileUnity Client Access to Read-Only for Configs

The ProfileUnity Client needs read-only access to the configuration storage account to manage users' desktops as they have been configured to deploy. Azure's Shared Access Signature (SAS) is a Uniform Resource Identifier (URI) that combines permission settings in the form of a token.

1. In the Azure portal, navigate back to the ProfileUnity Configuration Storage Account.
2. Under **Settings**, click on **Shared access signature**.
3. Configure the settings using the following guidance:

A shared access signature (SAS) is a URI that grants restricted access rights to Azure Storage resources. You can provide a shared access signature to clients who should not be trusted with your storage account key but whom you wish to delegate access to certain storage account resources. By distributing a shared access signature URI to these clients, you grant them access to a resource for a specified period of time.

An account-level SAS can delegate access to multiple storage services (i.e. blob, file, queue, table). Note that stored access policies are currently not supported for an account-level SAS.

[Learn more](#)

Allowed services ⓘ  
 Blob

Allowed resource types ⓘ  
 Service  Container  Object

Allowed permissions ⓘ  
 Read  Write  Delete  List  Add  Create  Update  Process

Start and expiry date/time ⓘ  
Start  
2018-04-16 2:08:46 PM  
End  
2023-04-16 10:08:46 PM  
(UTC-05:00) --- Current Timezone ---

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 ⓘ  
key1

[Generate SAS and connection string](#)

- a. Check the following:
    - i. Allowed services: **Blob**
    - ii. Allowed resource types: **Service, Container, Object**
    - iii. Allowed permissions: **Read, List**
  - b. Enter Start and End dates and times. We recommend your End date be 5 years or more from your Start date.
  - c. Click **Generate SAS and connection string** when done.
4. Azure will create 3 items: a connection string, a SAS token, and a Blob service SAS URL. Copy the **Blob service SAS URL** and save this information until you are ready to enter it into the ProfileUnity Management Console. In the ProfileUnity Management Console, the Blob service SAS URL for Microsoft Azure can be entered along with the Client Credentials in the Guided Configuration Wizard when using a cloud storage template. Or this can be entered in the Cloud Storage section of the Administration Settings when you click Copy next to Azure Client credentials.

## Putting This All Together

After completing all of these instructions to setup your Microsoft Azure Blob cloud storage for ProfileUnity, you will be able to install and configure ProfileUnity to make use of your new cloud storage accounts.

When using a cloud storage template, ProfileUnity's Guided Configuration Wizard will prompt you for your ProfileUnity Console Configuration credentials, your ProfileUnity Client Portability credentials and your Blob service SAS URL.

Please note that when ProfileUnity refers to Azure Blob cloud storage paths, they begin with "AZ://". Here are some examples where what is in brackets is replaced with the specified storage account name:

- **Deployment/Console Path:** AZ://{config-storage}
- **Portability/Client Path:** AZ://{portability-storage}
- **FlexApp Packages Path:** AZ://{config-storage}/flexapp
- **GPO Settings for INI, ProfileUnity as a Service, Client Settings XML Path:** AZ://{config-storage}

You may change your ProfileUnity Console or Client credentials at any time by going to the Cloud Storage Settings section in the ProfileUnity Management Console Administration Area.

Please see our ProfileUnity Installation and Configuration Guide for more instructions on how to adjust your Licensing and GPO configuration to utilize cloud storage.

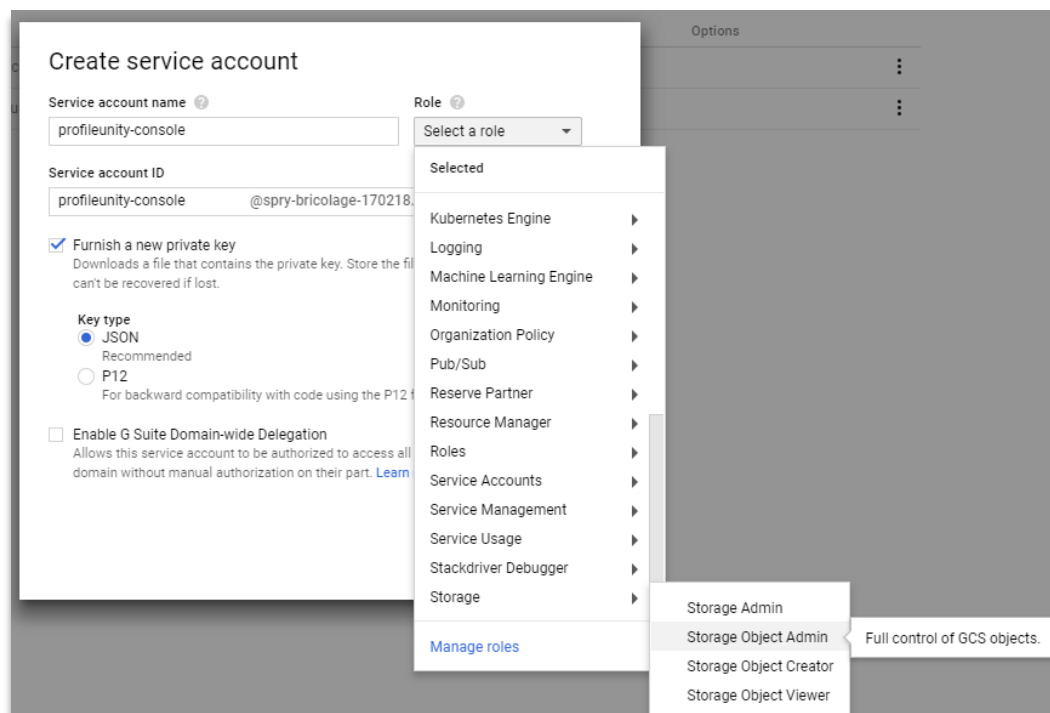
## Google Cloud Storage

Google Cloud Storage provides unified object storage for accessing data over the internet. The data is stored inside a resource called a “bucket”. Each bucket can hold as many objects as you want. Administrators control access to their storage bucket and who can read, write, or delete data objects in that bucket. For more detailed information, please visit the [Google Cloud Storage](#) website.

### Creating a Console Service Account

This is the Google Cloud Storage account that the ProfileUnity Management Console will use.

1. Login to your Google Cloud Platform.
2. Click on the triple bar button in the upper, left corner and select the **Permissions** page.
3. On the Permissions page, go to the **Service accounts** tab and click on the **Create service account** button.
4. Complete the service account information using the following guidance:



- a. Enter a **Service account name**. For example, “profileunity-console”.
- b. Select the **Role: Storage > Storage Object Admin**.
- c. Check the option to **Furnish a new private key**.
- d. Select **JSON** for the **Key type**.
- e. Click **Create** when done. A JSON file will be downloaded. Keep this file safe; you cannot download this key again. This key file will be used in the ProfileUnity Management Console.

## Creating a Client Service Account

This is the Google Cloud Storage account that the ProfileUnity Client will use.

1. Remain logged in to your Google Cloud Platform.
2. Click on the triple bar button in the upper, left corner and select the **Permissions** page.
3. On the Permissions page, go to the **Service accounts** tab and click on the **Create service account** button.
4. Complete the service account information using the following guidance:

**Create service account**

Service account name <sup>?</sup>  Role <sup>?</sup>

Service account ID

You don't have permission to furnish a new private key.

**Furnish a new private key**  
Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

**Key type**

**JSON**  
Recommended

**P12**  
For backward compatibility with code using the P12 format

You don't have permission to modify the domain-wide delegation setting You don't have permission to modify the product name for the consent screen

**Enable G Suite Domain-wide Delegation**  
Allows this service account to be authorized to access all users' data on a G Suite domain without manual authorization on their part. [Learn more](#)

**CANCEL CREATE**

- a. Enter a **Service account name**. For example, “profileunity-client”.
- b. Select the **Role: Storage > Storage Object Admin**.
- c. Check the option to **Furnish a new private key**.
- d. Select **JSON** for the **Key type**.
- e. Click **Create** when done. A JSON file will be downloaded. Keep this file safe; you cannot download this key again. This key file will be used in the ProfileUnity Management Console.

## Creating a Configuration Bucket

1. From the Google Cloud Platform, open the Cloud Storage Browser under **Storage > Browser** in the left navigation.
2. Click **Create bucket**.

3. Complete the bucket information using the following guidance:

← Create a bucket

**Name** ⓘ  
Must be unique across Cloud Storage. If you're [serving website content](#), enter the website domain as the name.

lw-profileunity-configs

**Default storage class** ⓘ  
[Compare storage classes](#)

Multi-Regional  
 Regional  
 Nearline  
 Coldline

**Location**

us-east1

<b>Storage cost</b>	<b>Retrieval cost</b>	<b>Class A operations</b> ⓘ	<b>Class B operations</b> ⓘ
\$0.02 per GB-month	Free	\$0.005 per 1,000 ops	\$0.0004 per 1,000 ops

⌵ Show advanced settings

Create Cancel

- a. Enter a bucket **name**. For example, “lw-profileunity-configs”. All bucket names must be unique. Therefore, the recommended name format is yourcompanyname-bucketuse-bucketfiles where “bucketuse” is what this particular bucket is being used for (in this case, ProfileUnity) and “bucketfiles” is the type of files that will be stored here, which in this case is your configuration files. Here are a few additional naming requirements:
  - i. Bucket names may contain lowercase letters, numbers, dashes, underscores, and dots.
  - ii. Bucket names can only start and end with a number or letter.
  - iii. Names must be between 3 and 63 characters long. Names containing dots can be up to 222 characters long with each dot-separated component being no longer than 63 characters.
  - iv. Names cannot be represented as an IP address.
  - v. Names may not begin with the “goog” prefix, contain “google” or contain close misspellings of “google”.
- b. Set the **Default storage class** to **Regional** and select the same location where your desktops are hosted. For more information, read about [Google’s Storage Classes](#).
- c. Click **Create** when done.

## Creating a Portability Bucket

1. From the Google Cloud Platform, open the Cloud Storage Browser under **Storage > Browser** in the left navigation.
2. Click **Create bucket**.

3. Complete the bucket information using the following guidance:

← Create a bucket

**Name** ⓘ  
Must be unique across Cloud Storage. If you're [serving website content](#), enter the website domain as the name.

lw-profileunity-portability

**Default storage class** ⓘ  
[Compare storage classes](#)

Multi-Regional  
 Regional  
 Nearline  
 Coldline

**Location**

us-east1

Storage cost	Retrieval cost	Class A operations ⓘ	Class B operations ⓘ
\$0.02 per GB-month	Free	\$0.005 per 1,000 ops	\$0.0004 per 1,000 ops

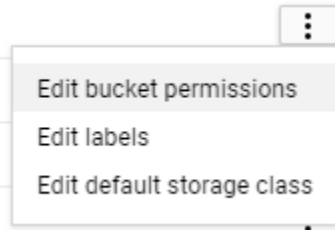
⌵ Show advanced settings

Create Cancel

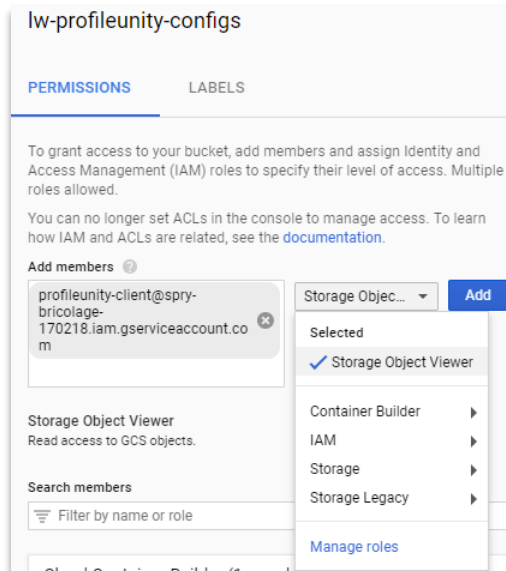
- a. Enter a bucket **name**. For example, “lw-profileunity-portability”. All bucket names must be unique. Therefore, the recommended name format is yourcompanyname-bucketuse-bucketfiles where “bucketuse” is what this particular bucket is being used for (in this case, ProfileUnity) and “bucketfiles” is the type of files that will be stored here, which in this case is your portability files. Here are a few additional naming requirements:
  - i. Bucket names may contain lowercase letters, numbers, dashes, underscores, and dots.
  - ii. Bucket names can only start and end with a number or letter.
  - iii. Names must be between 3 and 63 characters long. Names containing dots can be up to 222 characters long with each dot-separated component being no longer than 63 characters.
  - iv. Names cannot be represented as an IP address.
  - v. Names may not begin with the “goog” prefix, contain “google” or contain close misspellings of “google”.
- b. Set the **Default storage class** to **Regional** and select the same location where your desktops are hosted. For more information, read about [Google’s Storage Classes](#).
- c. Click **Create** when done.

## Setting Permissions on the Portability Bucket

1. From your list of buckets, find the ProfileUnity Configuration (“configs”) bucket. Click on the vertical ellipses to the right of the bucket name and select **Edit bucket permissions**.



2. Under the **Permissions** tab, add the ProfileUnity Client account as a **Storage Object Viewer**.



3. Click **Add** when done.

## Putting This All Together

After completing all of these instructions to setup your Google Cloud Storage for ProfileUnity, you will be able to install and configure ProfileUnity to make use of your new cloud storage accounts.

When using a cloud storage template, ProfileUnity’s Guided Configuration Wizard will prompt you for your ProfileUnity Console service account key and your ProfileUnity Client service account key. Copy and paste the contents of the JSON files that were previously downloaded when you created Google Cloud Storage service accounts.

Please note that when ProfileUnity refers to Google Cloud Storage paths, they begin with “GS://”. Here are some examples where what is in brackets is replaced with the specified storage bucket name:

- **Deployment/Console Path:** GS://{config-bucket}
- **Portability/Client Path:** GS://{portability-bucket}
- **FlexApp Packages Path:** GS://{config-bucket}/flexapp
- **GPO Settings for INI, ProfileUnity as a Service, Client Settings XML Path:** GS://{config-bucket}



You may change your ProfileUnity Console or Client credentials at any time by going to the Cloud Storage Settings section in the ProfileUnity Management Console Administration Area.

Please see our ProfileUnity Installation and Configuration Guide for more instructions on how to adjust your Licensing and GPO configuration to utilize cloud storage.

## Getting Help Installing ProfileUnity

If you have questions or run into issues while installing and configuring ProfileUnity with FlexApp, Liquidware is here to help. Our goal is to provide you with the knowledge, tools, and support you need to be productive.

### Using Online Resources

Liquidware maintains various kinds of helpful resources on our [Customer Support Portal](#). If you have questions about your product, please use these online resources to your full advantage. The Support Portal includes product forums, a searchable Knowledge Base, documentation, and best practices among other items. You can visit our website at <http://www.liquidware.com>.

### Contacting Support

If you wish to contact our Support staff for technical assistance, please either log a request on the [Liquidware Customer Support Portal](#) or give us a call. Prior to Logging a Case you may want to review these helpful tips:

- Check the Product Documentation included with your Liquidware Product.
- Try to see if the problem is reproducible.
- Check to see if the problem is isolated to one machine or more.
- Note any recent changes to your system and environment.
- Note the version of your Liquidware product and environment details such as operating system, virtualization platform version, etc.

To speak directly with Support, please use the following numbers:

<b>Main Line:</b>	1-678-397-0460
<b>Toll Free in US &amp; Canada:</b>	1-866-914-9665
<b>Europe/Middle East/Africa:</b>	+44 800 014 8097
<b>Toll Free in Europe</b>	
<b>UK:</b>	0800 014 8097
<b>Netherlands:</b>	0800 022 5973
<b>Switzerland:</b>	0800 561 271