Getting started with blob storage in .NET

Mary Loubele, PhD
Senior Data Scientist FunnelCake

Introduction

- Needed resources for the exercises
- What is blob storage
- Use case
- How to develop with blob storage
  - Azure portal
  - Azure Storage emulator
- Exercise
  - Building the use case
- Questions

What do we need before hand

- Visual Studio installed. Free version is Visual Studio Community https://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx
- An instance of SQL Server installed SQL Server 2016 Express is a free option https://www.microsoft.com/en-gb/cloud-platform/sql-server-editions-express
- Stand alone installer from here https://azure.microsoft.com/en-us/documentation/articles/storage-use-emulator/
- https://azure.microsoft.com/en-us/documentation/articles/storage-dotnet-how-to-use-blobs/ (useful link)

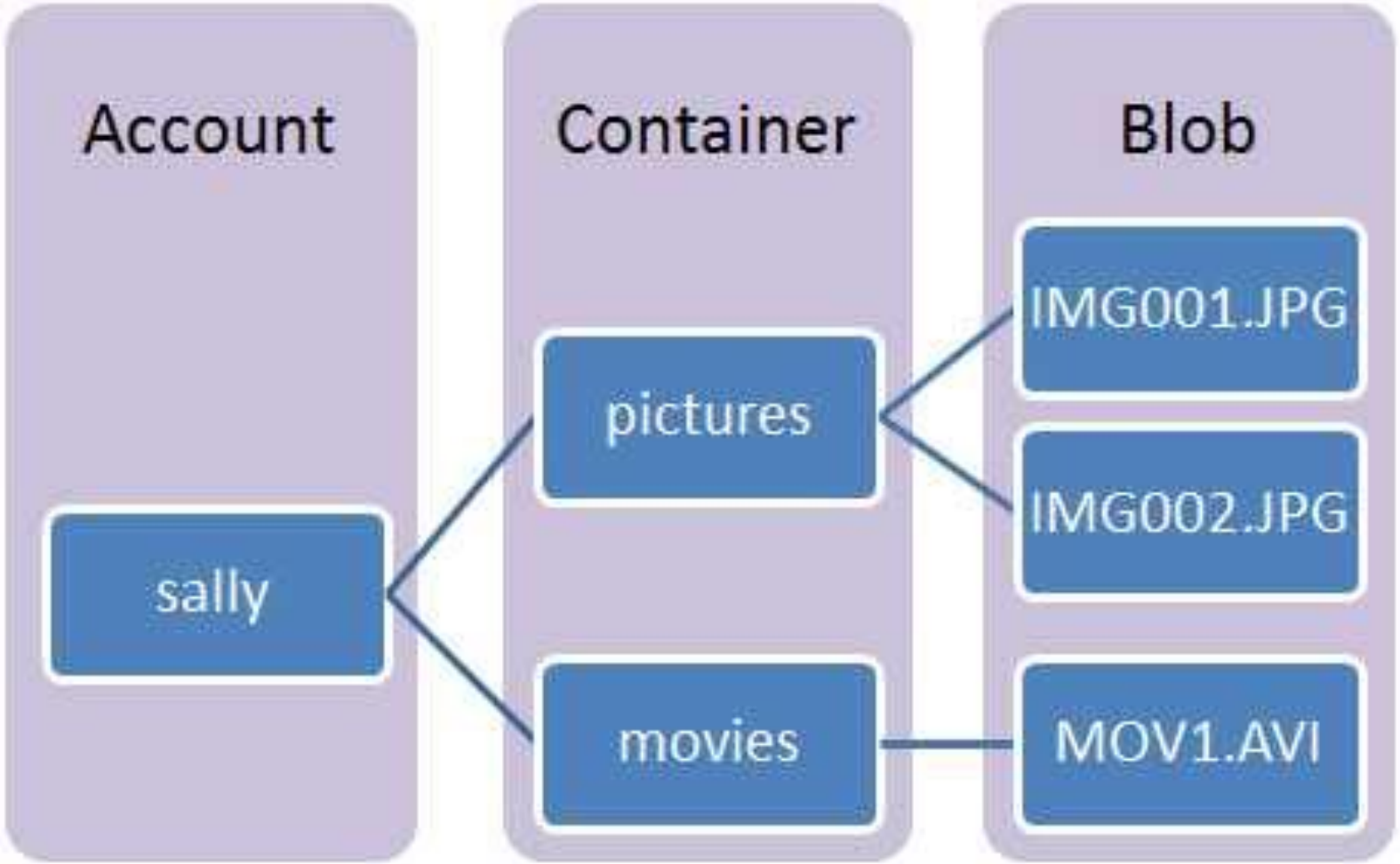
Introduction

- Needed resources for the exercises
- What is blob storage
- Use case
- How to develop with blob storage
  - Azure portal
  - Azure Storage emulator
- Exercise
  - Building the use case
- Questions

What is Blob Storage

- Service for storing large amounts of unstructured object data
  - Text or binary
  - Accessible through http or https
  - Publicly to the world or to store application data
- Common use
  - Serving images or documents directly to a browser
  - Storing files for distributed access
  - Streaming video and audio
  - Storing data for backup and restore, disaster recovery and archiving
  - Storing data for analysis by an on-premise or Azure-hosted service

Blob service concepts



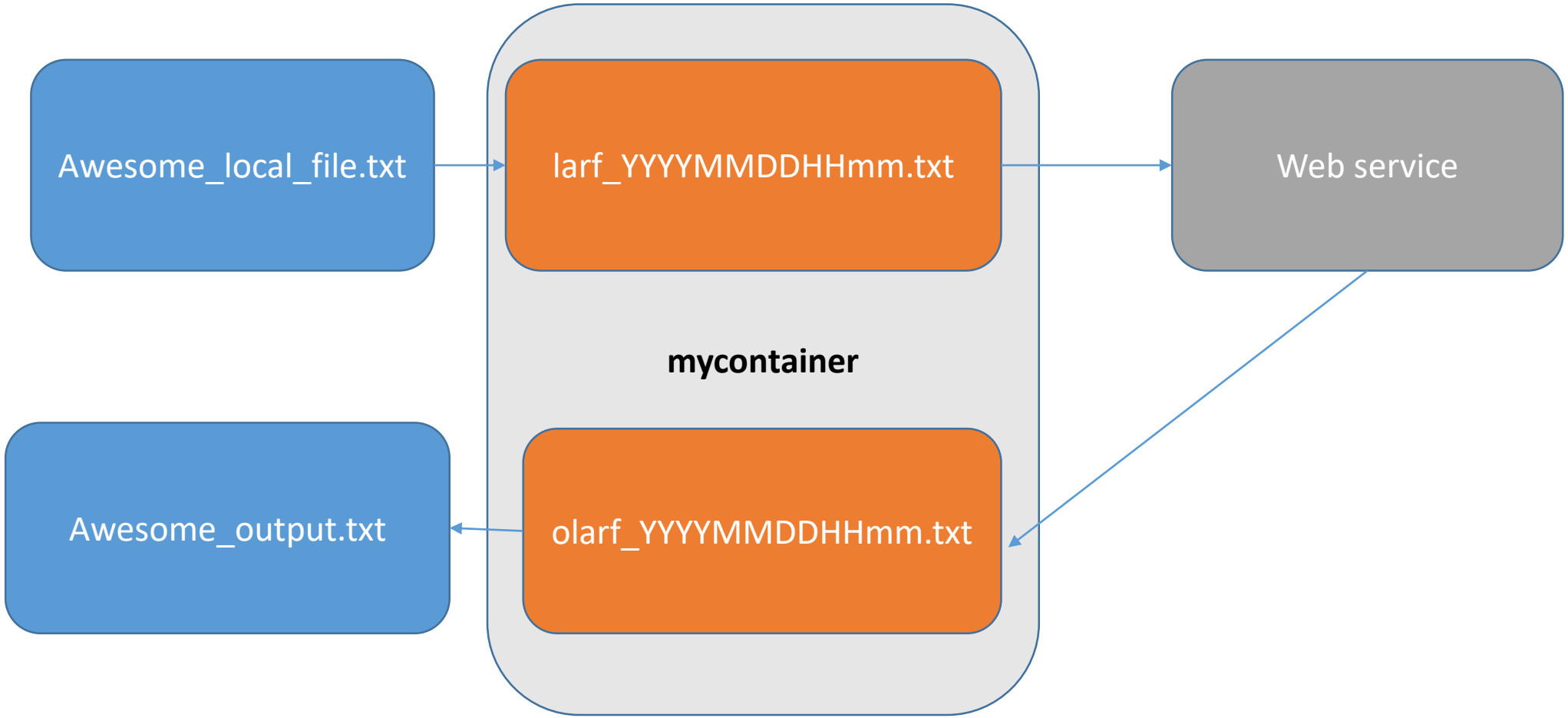
https://myaccount.blob.core.windows.net/mycontainer/myblob

Introduction

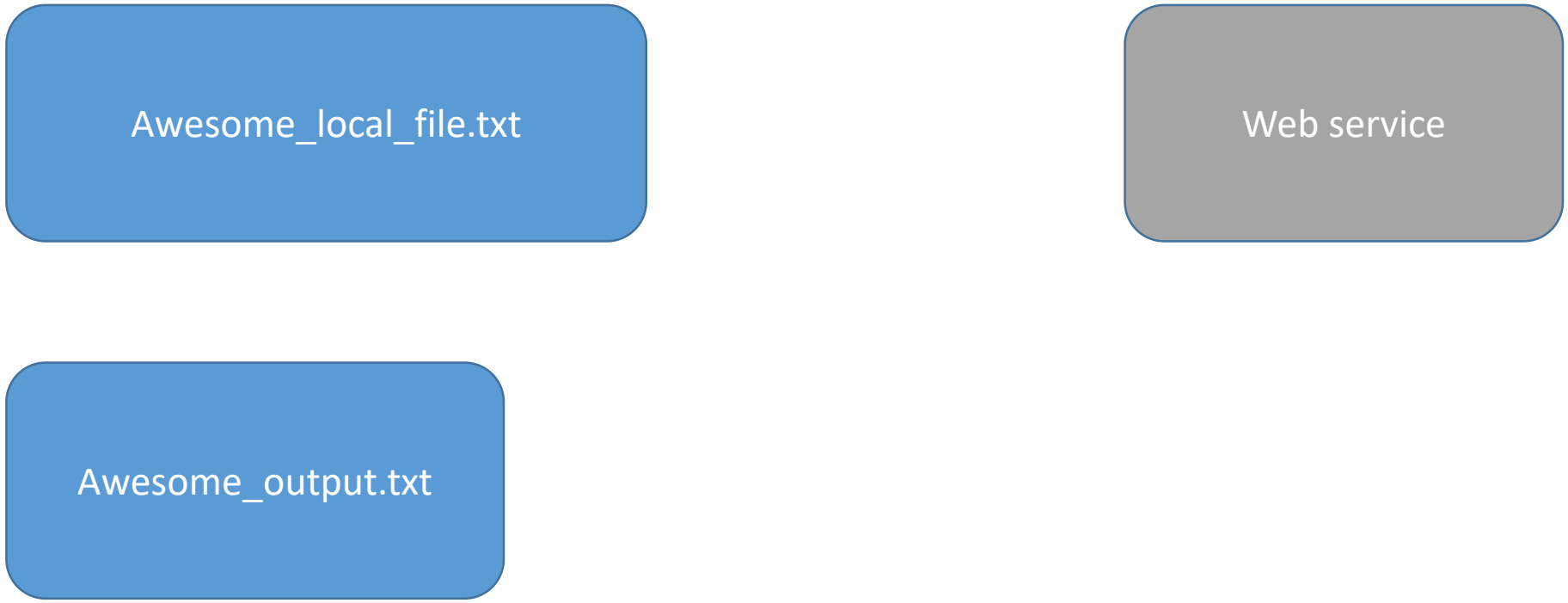
- Needed resources for the exercises
- What is blob storage
- Use case
- How to develop with blob storage
  - Azure portal
  - Azure Storage emulator
- Exercise
  - Building the use case
- Questions

Our application (0)

Our application (1)



Our application (2) Cleaning everything up

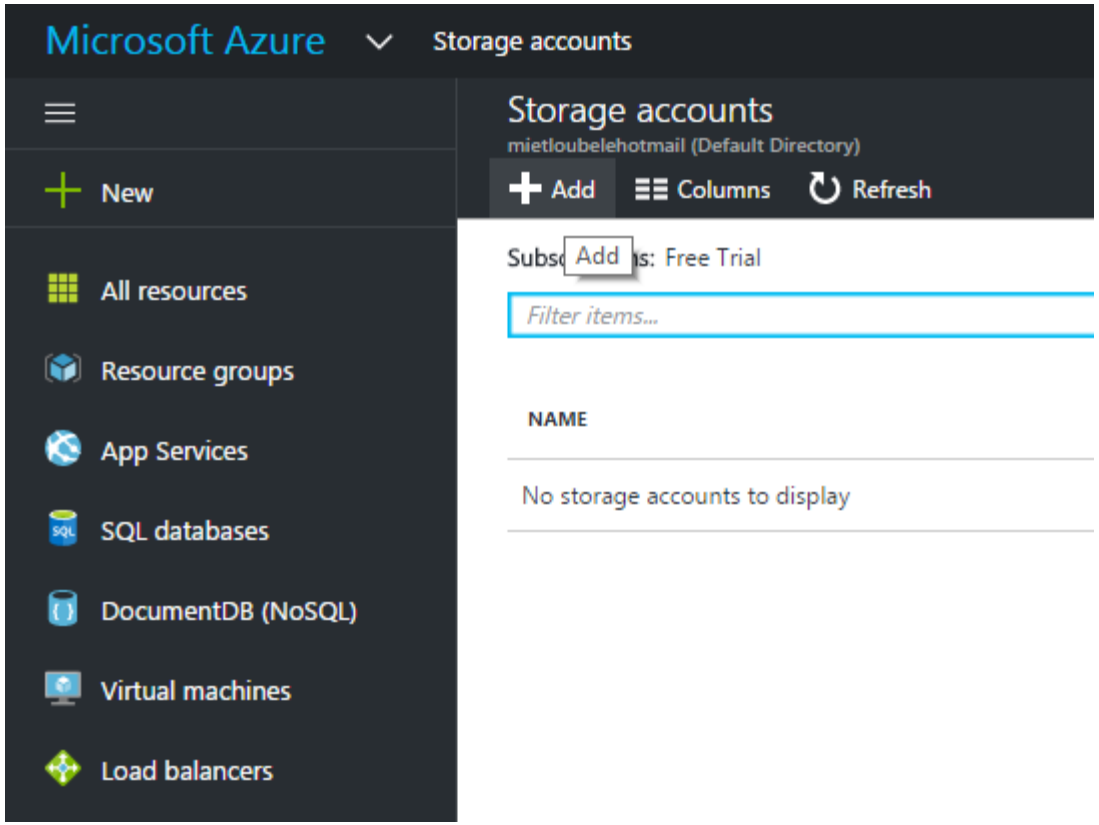


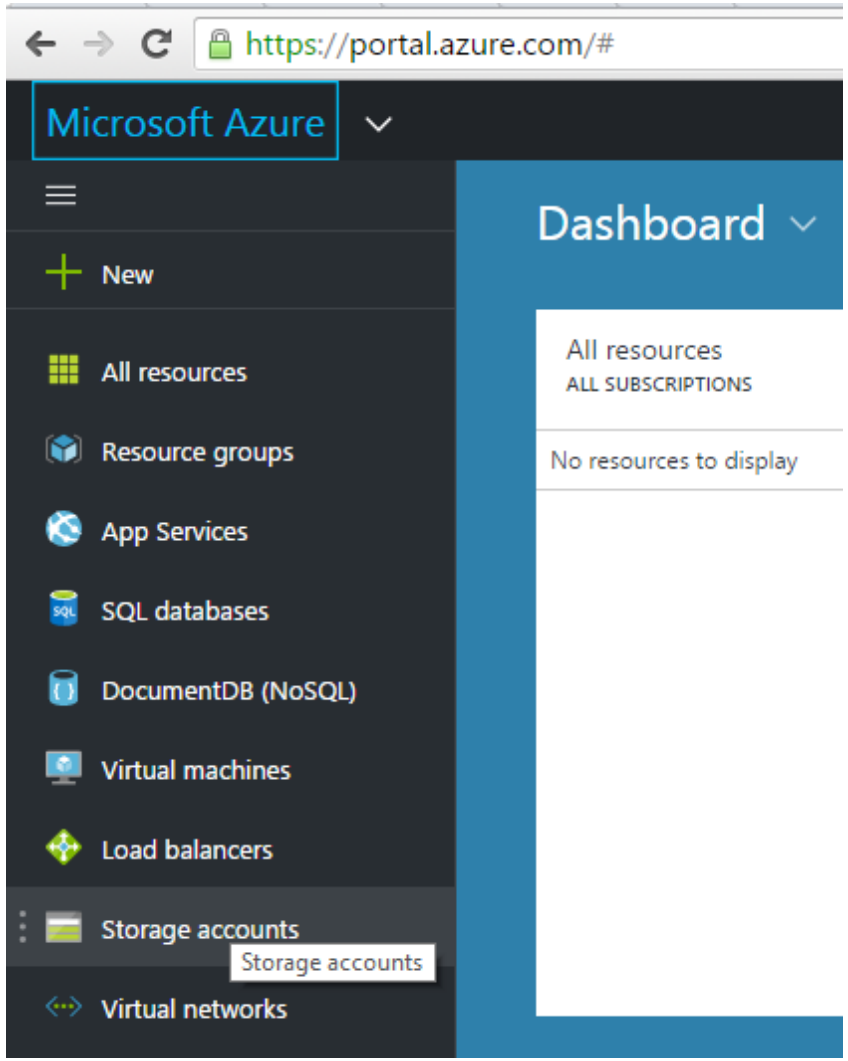
Introduction

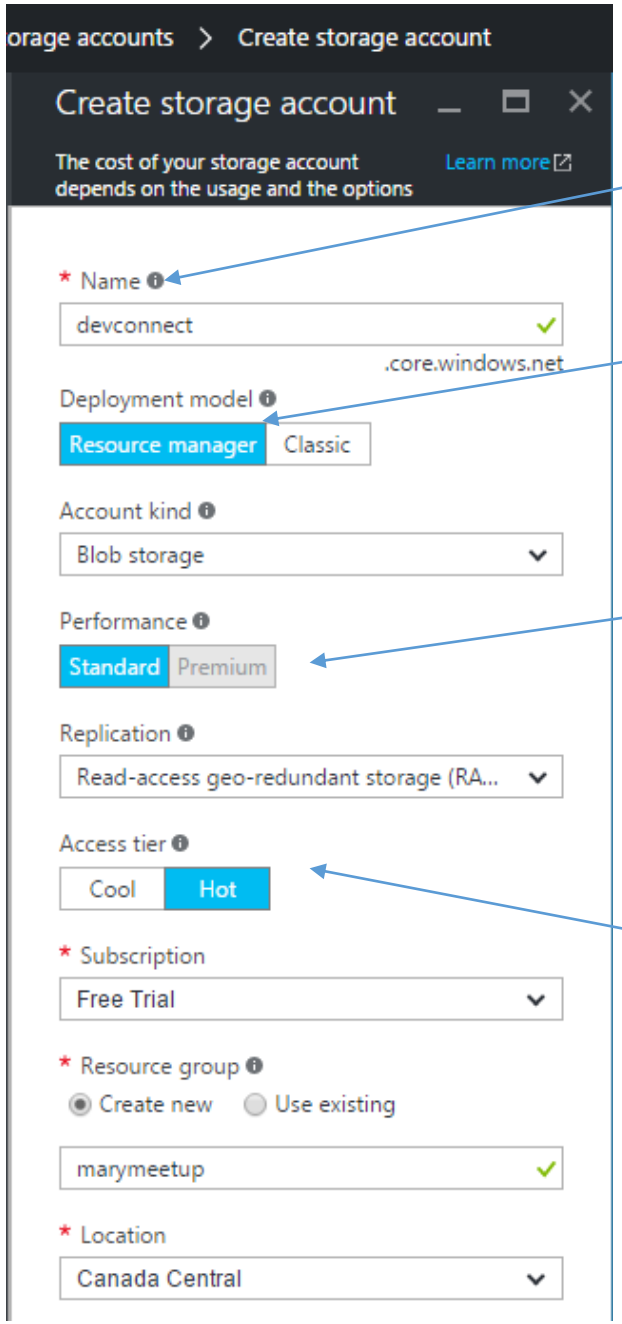
- Needed resources for the exercises
- What is blob storage
- Use case
- How to develop with blob storage
  - Azure portal
  - Azure Storage emulator
- Exercise
  - Building the use case
- Questions

Create an Azure storage Account

- Sign up for a free azure trial account
- Create a blob storage account







Unique over all azure blob storage, 3 to 24 characters, only numbers and lowercase

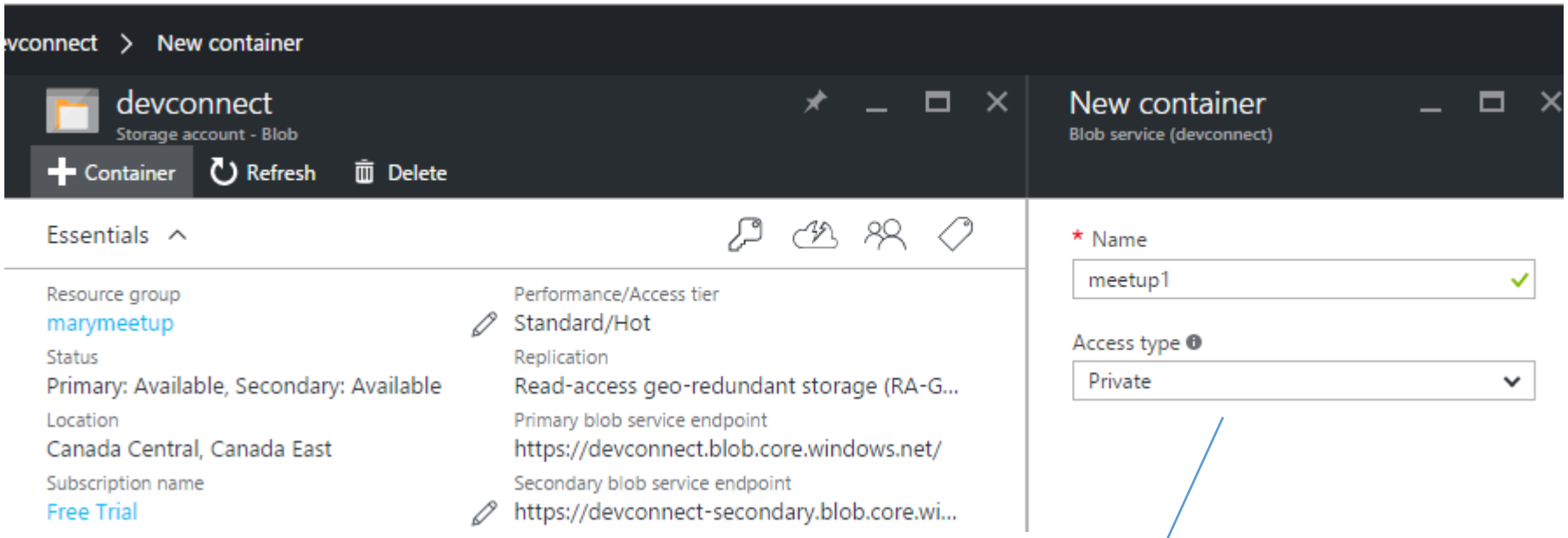
Resource manager for new applications, classic for old

https://azure.microsoft.com/en-us/documentation/articles/resource-manager-deployment-model/

Standard storage backed by magnetic drives, lowest cost per GB
Premium storage backed by solid state drives and offer low-latency performance, only with Azure virtual machine discs

Cool optimized for backup
Hot optimized for frequently accessed data

Generation of Access Keys and new container



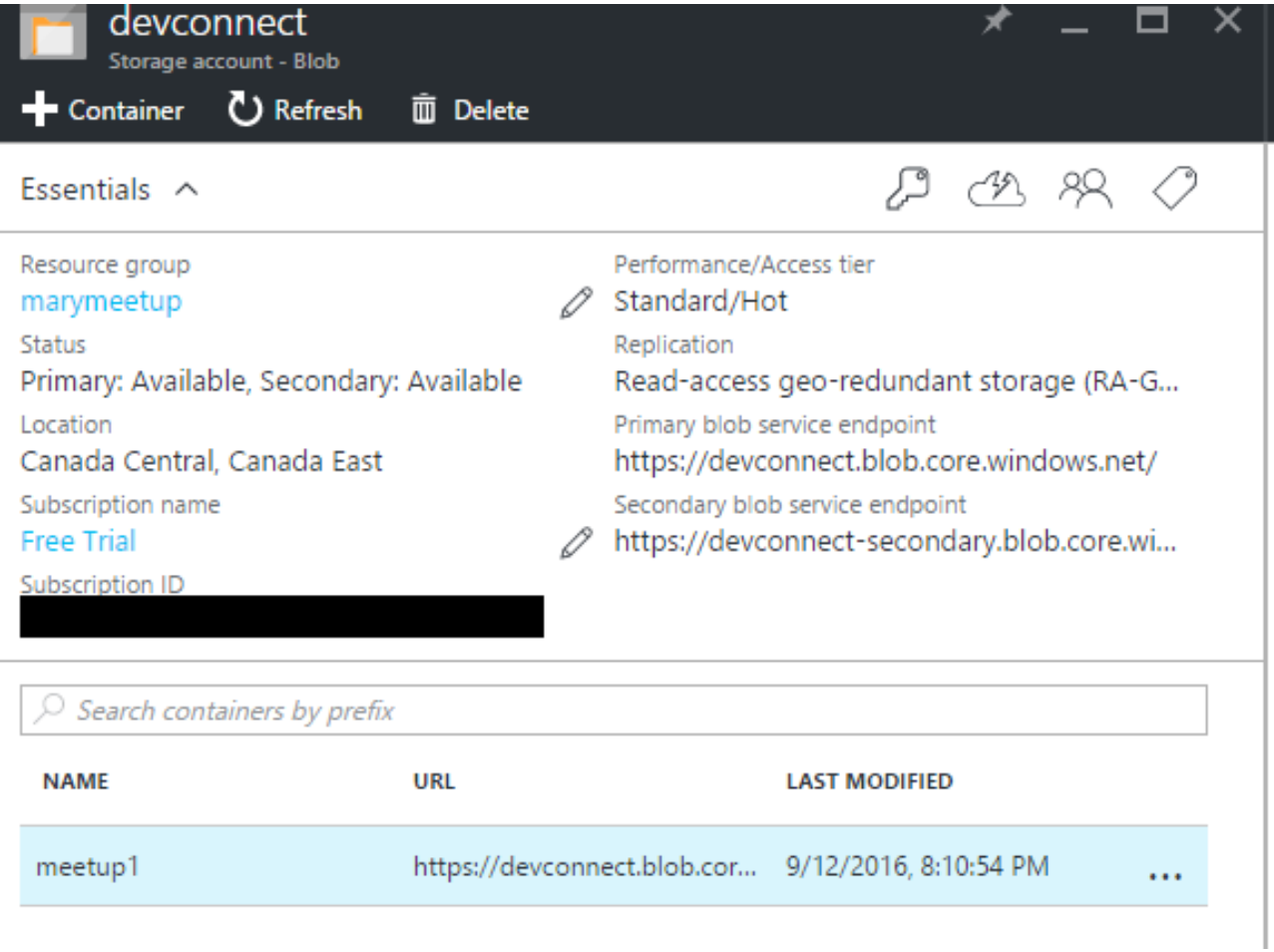
Specifies whether data can be accessed publicly

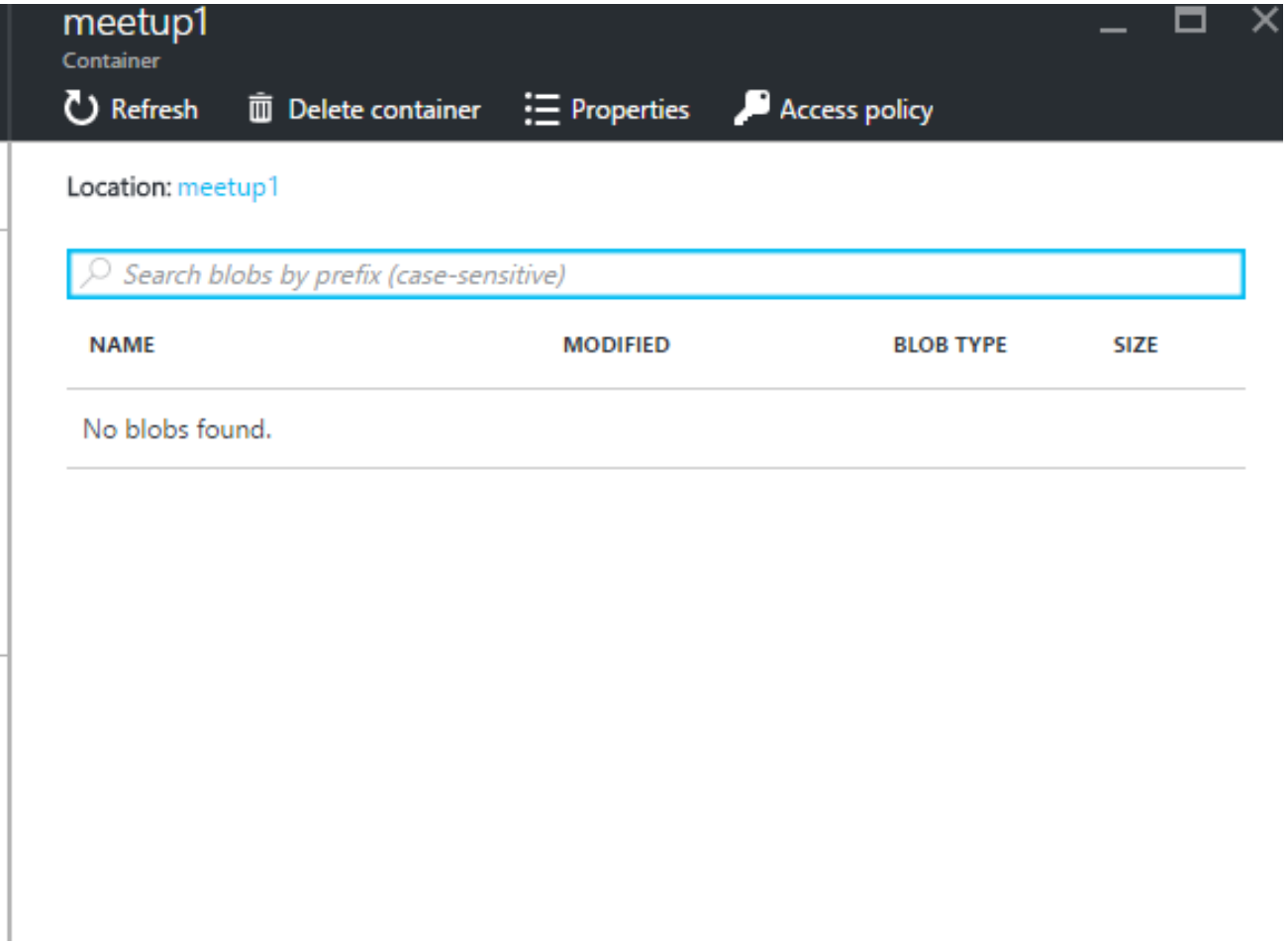
Container data is **private** to the account owner

'**Blob**' to allow public read access for blobs

'**Container**' to allow public read and list access to the entire container

Resulting container





Intermezzo 1: Azure Storage Emulator

- Development without immediate access to the Azure cloud
- Install Stand alone installer https://azure.microsoft.com/en-us/documentation/articles/storage-use-emulator/
- Start **Azure Storage Emulator** from “Search Windows”

C:\WINDOWS\system32\cmd.exe

```
C:\Program Files (x86)\Microsoft SDKs\Azure\Storage Emulator>AzureStorageEmulator.exe start
Windows Azure Storage Emulator 4.4.0.0 command line tool
The storage emulator was successfully started.

C:\Program Files (x86)\Microsoft SDKs\Azure\Storage Emulator>cmd /K AzureStorageEmulator.exe help
Windows Azure Storage Emulator 4.4.0.0 command line tool
Usage:
    AzureStorageEmulator.exe init            : Initialize the emulator database and configuration.
    AzureStorageEmulator.exe start           : Start the emulator.
    AzureStorageEmulator.exe stop            : Stop the emulator.
    AzureStorageEmulator.exe status          : Get current emulator status.
    AzureStorageEmulator.exe clear           : Delete all data in the emulator.
    AzureStorageEmulator.exe help [command]  : Show general or command-specific help.

See the following URL for more command line help: http://go.microsoft.com/fwlink/?LinkId=392235

C:\Program Files (x86)\Microsoft SDKs\Azure\Storage Emulator>
```

Authenticating requests against the storage emulator

Account name: devstoreaccount1
Account key: Eby8vdM02xNOcqFlqUwJPLlmEtlCDXJ1OUzFT50uSRZ6IFsuFq2UVErCz4I6tq/K1SZFPTOtr/KBHBeksoGMGw==

```
In app.config file

<appSettings>
        <add key="StorageConnectionString" value="UseDevelopmentStorage=true;" />
</appSettings>
```

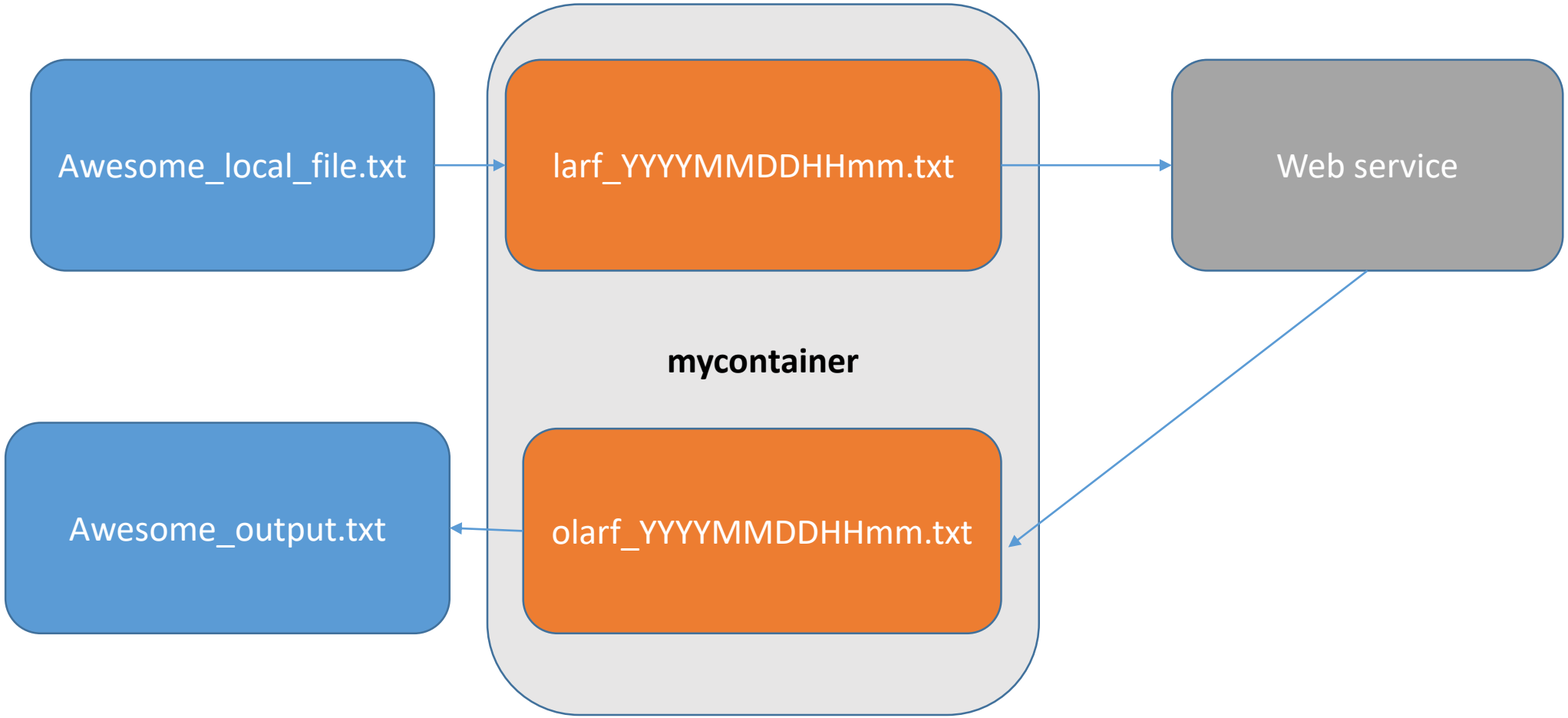
```
DefaultEndpointsProtocol=http;AccountName=devstoreaccount1;
AccountKey=Eby8vdM02xNOcqFlqUwJPLlmEtlCDXJ1OUzFT50uSRZ6IFsuFq2UVErCz4I6tq/K1SZFPTOt
BlobEndpoint=http://127.0.0.1:10000/devstoreaccount1;
TableEndpoint=http://127.0.0.1:10002/devstoreaccount1;
QueueEndpoint=http://127.0.0.1:10001/devstoreaccount1;
```

Exercise: starting the emulator

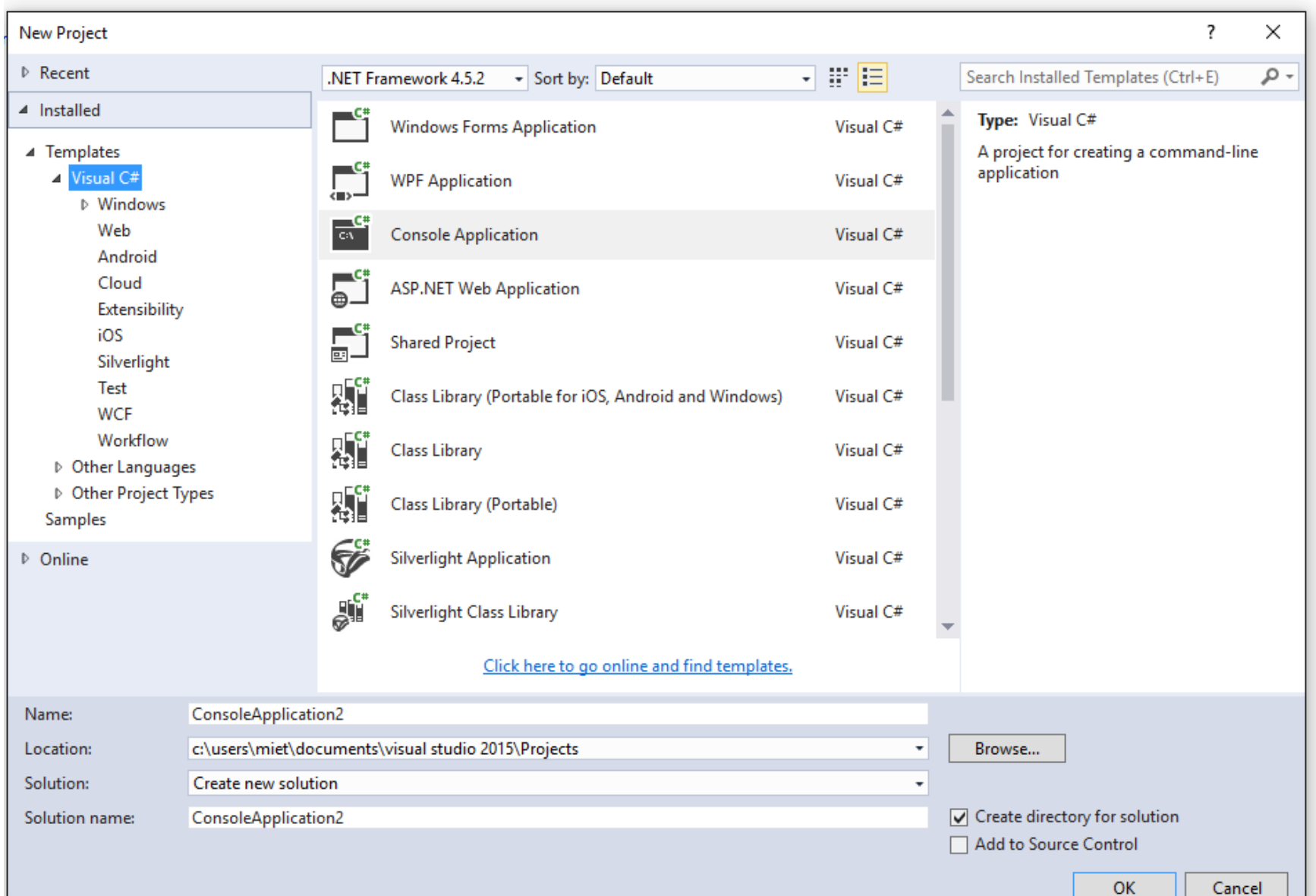
Introduction

- Needed resources for the exercises
- What is blob storage
- Use case
- How to develop with blob storage
  - Azure portal
  - Azure Storage emulator
- Exercise
  - Building the use case
- Questions

Use case


Awesome_local_file.txt
larf_YYYYMMDDHHmm.txt
Web service
mycontainer
Awesome_output.txt
olarf_YYYYMMDDHHmm.txt

Exercise 0: Setting up the development environment



Start Page - Microsoft Visual Studio

File Edit View Debug Team Tools Test Analyze Window Help

1

- Connect to Database...
- Connect to Server...
- SQL Server
- Code Snippets Manager... Ctrl+K, Ctrl+B
- Choose Toolbox Items...
- NuGet Package Manager
  - Package Manager Console
  - Package Manager Settings
- Extensions and Updates...
- Create GUID
- PreEmptive Protection - Dotfuscator
- WCF Service Configuration Editor
- External Tools...
- Import and Export Settings...
- Customize...
- Options...

2

Start Page

Start

New Project...

Open Project...

Open from Source Control...

Recent

C# 7.0

language features in C# 7.0. With the release of Visual Studio "15" Preview 4, most of these features are com...

NEW Thursday, August 25, 2016

Announcing the "App Dev on Xbox" live event (UWP on Xbox!)

Windows 10 Anniversary Update was released earlier this month and became instantly available to all existing Windows 10 devices, including PCs, phones,...

NEW Tuesday, August 23, 2016

The week in .NET – 8/23/2016

Last week, we moved into our new studios with Channel 9, where JB Evain joined me to interview Lucas Meijer from Unity. We had our fair share of technical...

NEW Thursday, August 25, 2016

Introducing the UWP Community Toolkit

Recently, we released the Windows Anniversary Update and a new Windows Software Developer Kit (SDK) for Windows 10 containing tools, app templates, platfor...

NEW Tuesday, August 23, 2016

New navigation for Team Services (Brian Harry)

In my blog post announcing TFS 15 RC1, I mentioned that we are working on a new navigation UI. In a somewhat unusual turn of events, it showed up in an...

NEW Thursday, August 25, 2016

Notes from the ASP.NET Community Standup Aug. 2016

This is the next in a series of blog posts that will cover the topics discussed in the ASP.NET Community Standup. The community standup is a short video-ba...

NEW Tuesday, August 23, 2016

Live from LinuxCon – Sharing the latest news and learnings on Microsoft's open journey

Greetings from LinuxCon North America in Toronto, where I am representing Microsoft as a keynote speaker

Portal support for Azure Search blob and table indexers now in preview

When building a search enabled application, data can come from many places and take many forms, so making it easy to ingest a variety of data sources is ex...

New on Microsoft Platforms

- Windows
- Microsoft Azure
- ASP.NET and Web

Featured Videos

- NEW IntelliTrace Experience 12:13
- NEW Windows Presentation Foundation Application Develop... 15:49
- NEW What's New in C# 6.0 7:49
- NEW What's New for .NET 2015 8:21
- NEW Code Editor Enhancements for XAML Developers 4:08

More videos

Package Manager Console

Package source: All Default project:

```
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages. Some packages may include dependencies which are governed by additional licenses. Follow the package source (feed) URL to determine any dependencies.

Package Manager Console Host Version 3.4.4.1321

Type 'get-help NuGet' to see all available NuGet commands.

PM> Install-Package WindowsAzure.Storage
```

3

Installing some packages and configuring app.config

- Install-Package WindowsAzure.Storage
- Install-Package Microsoft.WindowsAzure.ConfigurationManager

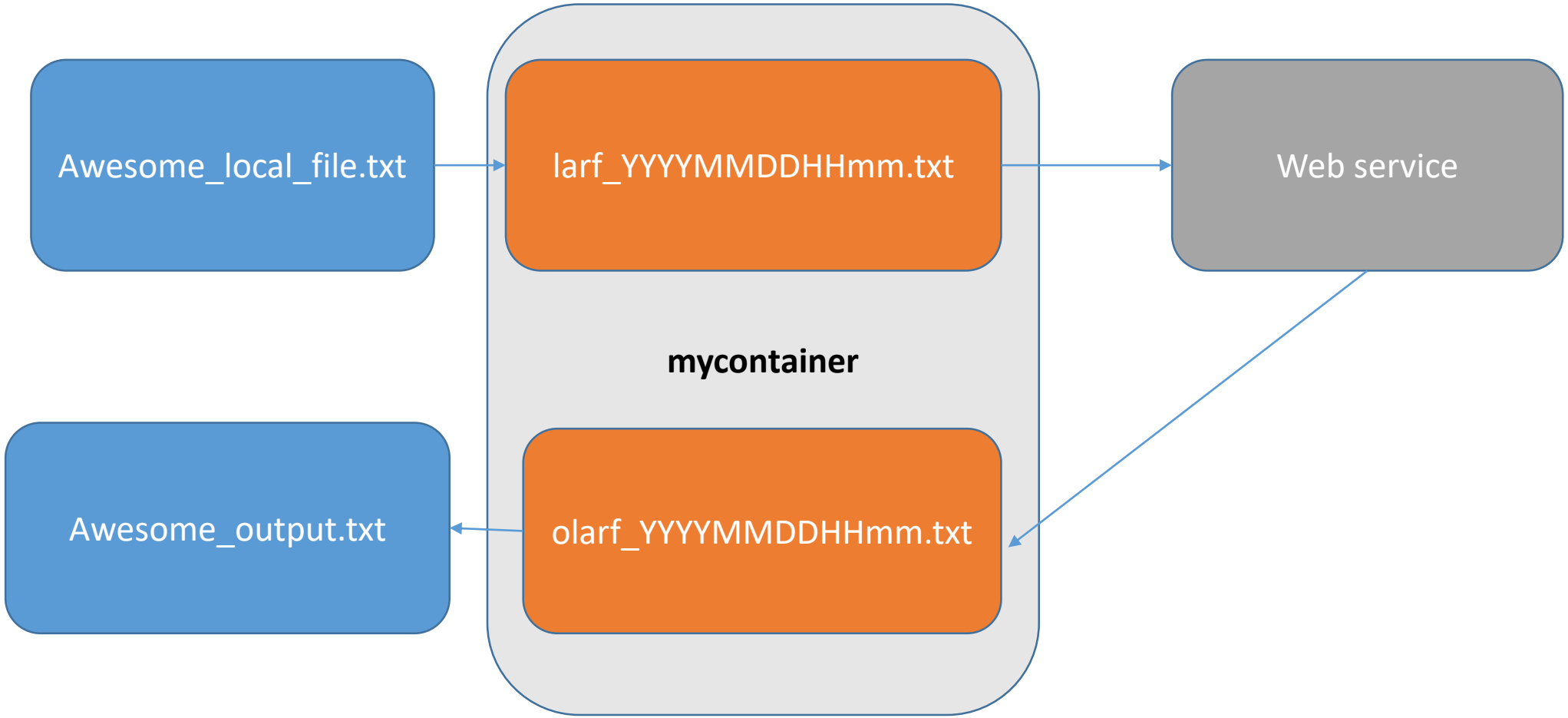
```
In app.config file

<appSettings>
        <add key="StorageConnectionString" value="UseDevelopmentStorage=true" />
</appSettings>
```

Structure of the exercises

- Make a folder test_files with the file Awesome_local_file.txt in it
- Each exercise
  - `public static void Exercise_X() (except for exercise 1 String)`
- Main method

```
static void Main(string[] args)
        {
            Console.WriteLine("Starting exercise.");
            Exercise_0();
            Console.WriteLine("Ending exercise.");
            Console.ReadKey();

        }
```

Exercise 0 accessing the storage account



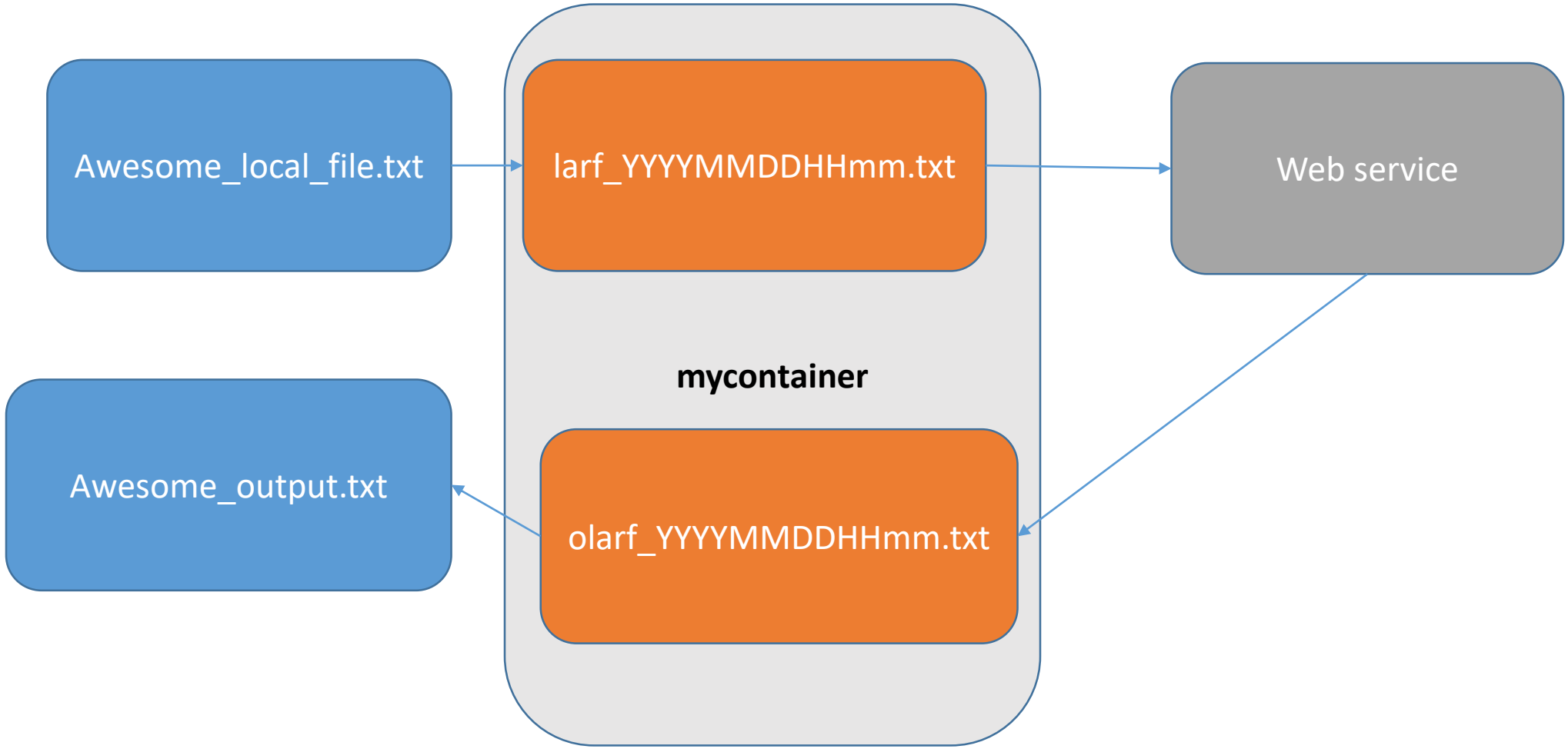
Used libraries

```
using Microsoft.Azure; // Namespace for CloudConfigurationManager
using Microsoft.WindowsAzure.Storage; // Namespace for CloudStorageAccount
using Microsoft.WindowsAzure.Storage.Blob; // Namespace for Blob storage types
```

Exercise 0: Access the storage account

```
public static void Exercise_0()
        {
            // Parse the connection string and return a reference to the storage account.
            CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
                CloudConfigurationManager.GetSetting("StorageConnectionString"));

            CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();

            // Retrieve a reference to a container.
            CloudBlobContainer container =
blobClient.GetContainerReference("mycontainer");

            // Create the container if it doesn't already exist.
            container.CreateIfNotExists();
            Console.WriteLine("Created successfully the container if it was not there
yet");
        }
```

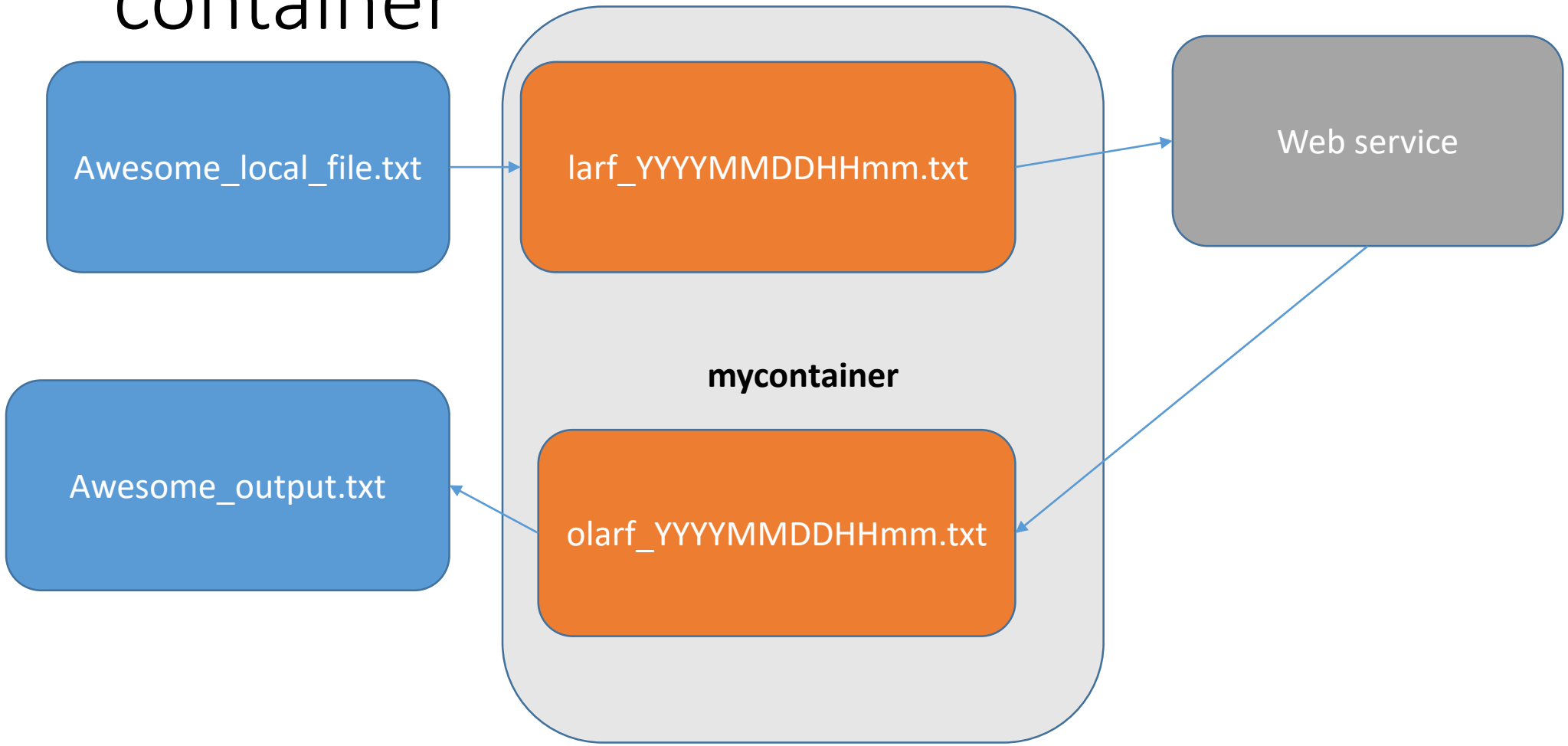
Exercise_1: copy local file to the cloud



Exercise_1: copy local file to the cloud

```
 // Retrieve reference to a blob named "myblob".
            String timestamp =
DateTime.Now.ToString("yyyyMMddHHmmssFFF");
            String blockName = "larf_" + timestamp;
            Console.WriteLine("We will generate the following file
" + blockName);
            CloudBlockBlob blockBlob =
container.GetBlockBlobReference(blockName);

            using (var fileStream =
System.IO.File.OpenRead(@"C:\test_files\Awesome_local_file.txt"))
            {
                blockBlob.UploadFromStream(fileStream);
            }

            return blockName;
```

Exercise 2: List all the blob files in the container



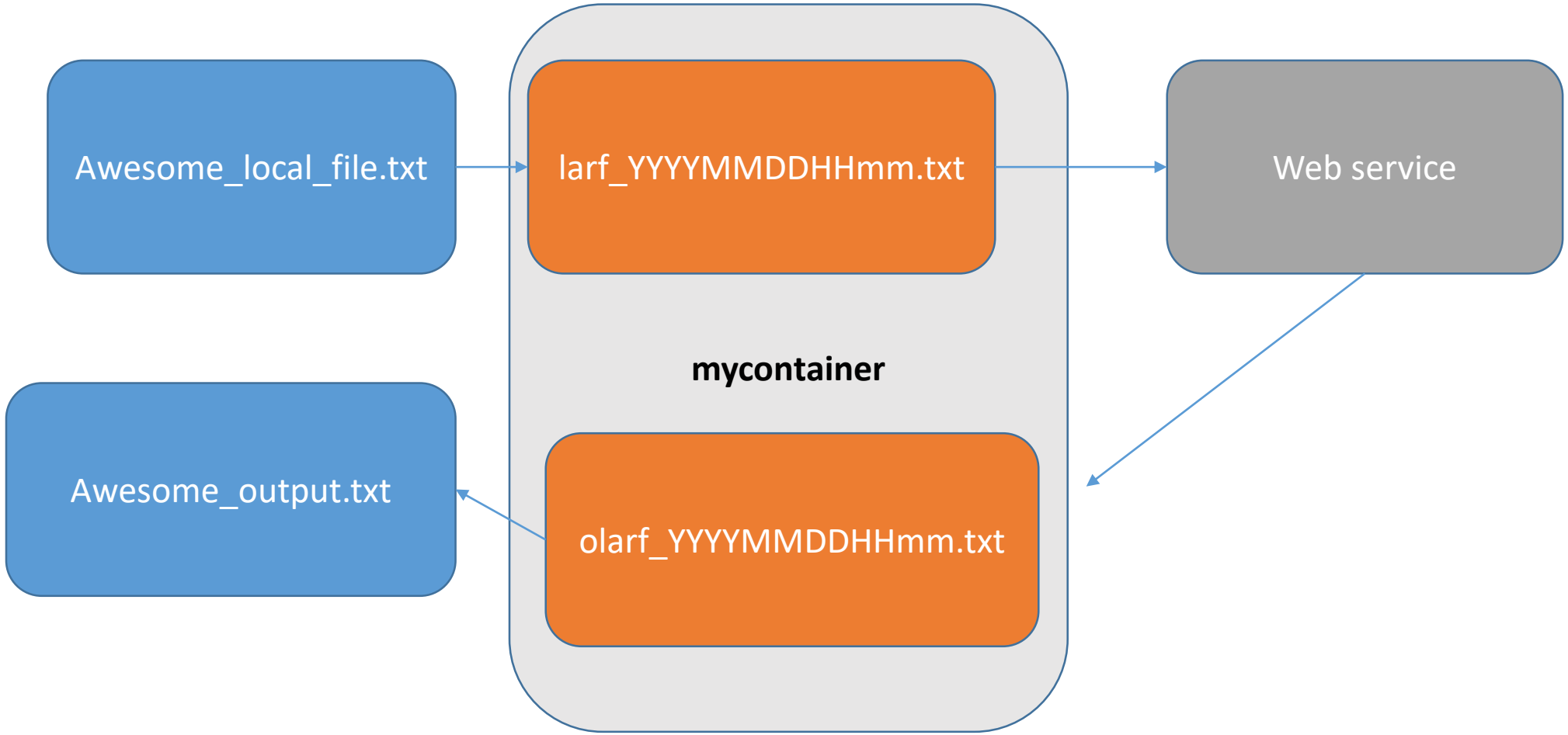
Exercise 2: List all the blob files in the container

```
 CloudStorageAccount storageAccount =
CloudStorageAccount.Parse(

CloudConfigurationManager.GetSetting("StorageConnectionSt
ring"));


            CloudBlobClient blobClient =
storageAccount.CreateCloudBlobClient();


            CloudBlobContainer container =
blobClient.GetContainerReference("mycontainer");
```

Exercise_2 part 2

```
foreach (IListBlobItem item in container.ListBlobs(null, false))
            {
                if (item.GetType() == typeof(CloudBlockBlob))
                {
                    CloudBlockBlob blob = (CloudBlockBlob)item;

                    Console.WriteLine("Block blob of length {0}: {1}",
blob.Properties.Length, blob.Uri);

                }
            }
```

Exercise 3: Copy blob and download new blob



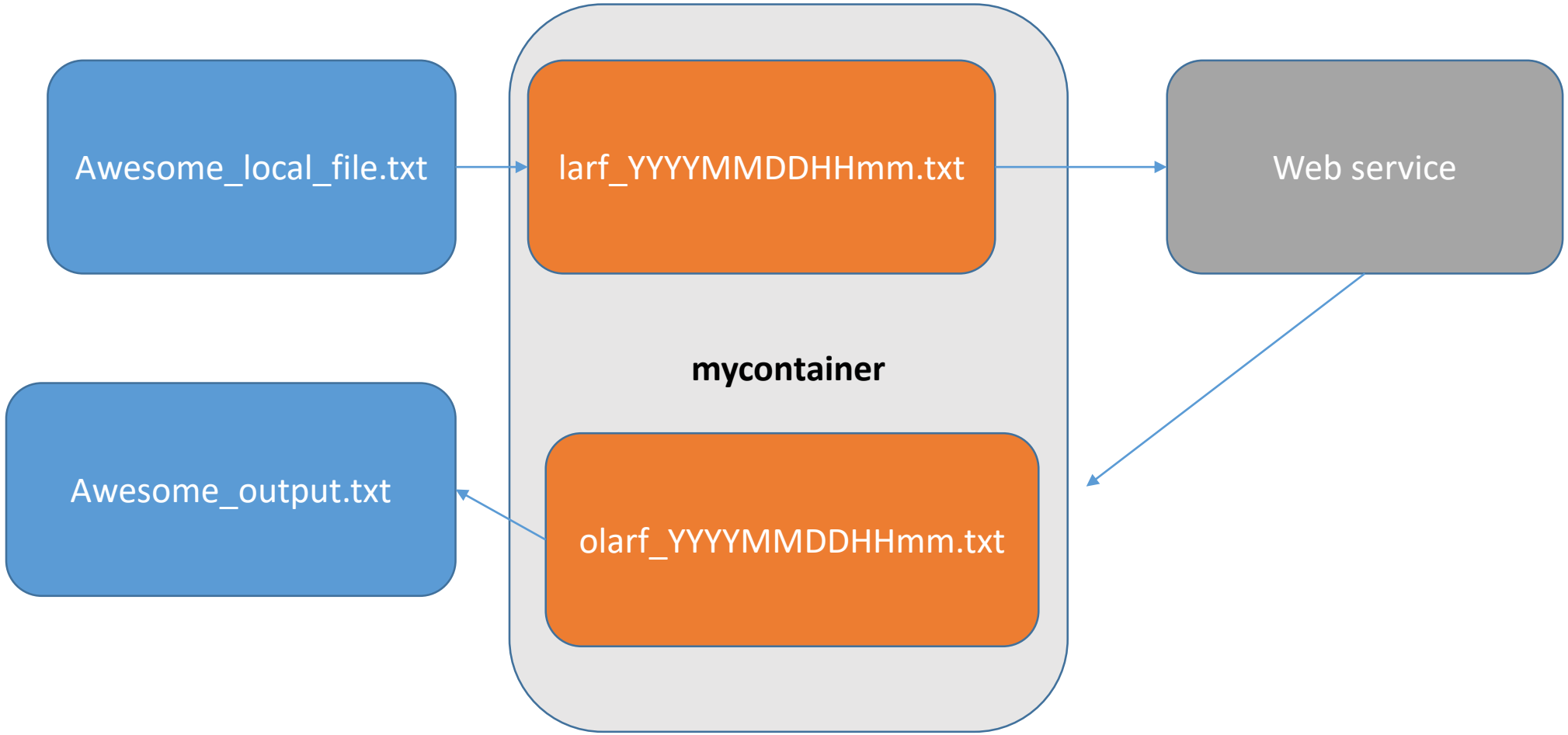
Exercise 3: Copy blob and download new blob

```
String blockName = Exercise_1();
            String blockOutputName = "o" + blockName;
            // Retrieve storage account from connection string.
            CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
                CloudConfigurationManager.GetSetting("StorageConnectionString"));

            // Create the blob client.
            CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();

            // Retrieve reference to a previously created container.
            CloudBlobContainer container =
blobClient.GetContainerReference("mycontainer");
```

Exercise 3: Download blobs

```
            CloudBlockBlob blockBlob =
container.GetBlockBlobReference(blockName);
            CloudBlockBlob targetBlob =
container.GetBlockBlobReference(blockOutputName);
            targetBlob.StartCopy(blockBlob);

            // Save blob contents to a file.
            using (var fileStream =
System.IO.File.OpenWrite(@"C:\test_files\AwesomeOutput.txt"))
            {
                targetBlob.DownloadToStream(fileStream);
            }
```

Exercise 4: cleaning up everything nicely



Exercise_4: cleaning up everything nicely

Add the following to exercise_3

```
 targetBlob.Delete();
blockBlob.Delete();
```

Exercise 5: Clean everything what is left

Make a function that cleans the remaining files and the container that you made, so combining what you learned in exercises 2 and 4

Questions