Представим такую бизнес задачу: предоставить клиентам возможность хранить данные, архивы, бэкапы в облаке. Причем, у вас уже есть приложение, которое используют клиенты, и это приложение позволяет идентифицировать пользователей. Список пользователей хранится на вашем сервере. Сервер, также, выполняет регистрацию и логирование пользователей. Т.е. вы хотите сохранить существующий механизм sign up/sign in, и добавить приложению функциональность хранения бэкап-данных в персональном пользовательском хранилище в облаке.

Один из вариантов решения может быть использование хранилища BLOB-объектов Azure.

Принципиальная схема работы будет следующая:   
- клиентское приложение регистрирует пользователя на вашем сервере  
- сервер формирует специальный токен и передает его клиентскому приложению. Токен позволяет доступ, с определенными правами и на определенное время, к контейнеру пользователя в облаке . В терминах Azure этот токен называется SAS (shared access signatures).  
- клиентское приложение, используя токен, получает доступ к своему контейнеру и сохраняет в контейнере или читает из него пользовательские данные в BLOB-объектах.

Разделение ресурсов, в данном случае будет таким:  
1. Ваш сервер имеет учетную запись (storage account) Azure.  
2. Под этой учетной записью он управляет (создает, удаляет, предоставляет доступ) контейнерами (container) пользователей.  
3. Пользовательское приложение записывает, читает, удаляет BLOB-объекты в пределах своего контейнера.

Хранилище BLOB-объектов Azure поддерживает несколько типов BLOB-объектов. В нашем примере мы будем использовать блочные BLOB (block blob).

В нашем простом примере мы используем эмулятор хранилища Azure (Azure storage emulator). Это позволит протестировать приложение локально, без использования подписки Azure. Но ничто не мешает настроить приложение для работы в настоящем облаке. Описание эмулятора:   
<https://docs.microsoft.com/en-us/azure/storage/common/storage-use-emulator>

Также рекомендуется использовать Microsoft Azure Storage Explorer для просмотра и управления хранилищем.

Итак, убедитесь, что установленны:  
Visual Studio 2017  
.NET Core SDK (last version)  
Microsoft Azure SDK  
Azure storage emulator (он входит в состав SDK, но иногда полезно обновить)  
Microsoft Azure Storage Explorer

Откройте BlobStore.sln в студии. Ребилд. Запустите проект BlobStore. Не останавливая его, запустите проект Client. (Можно запустить две студии, в каждой запустить свой проект). В методе Do(), класса Program, проекта Client наблюдаем взаимодействие клиентского приложения с хранилищем Azure.  
Также можно запустить Azure Storage Explorer и увидеть созданый контейнер пользователя и в нем Blob-объект.