KAZAKH KBTU BRITISH
TECHNICAL
UNIVERSITY

Cloud Development, 2021 fall.
Homework 2

Report

Storages

made by Zhuman Rakhat

Almaty 2021

Task

You have to develop a simple web application (one-page) which will display a list of files (images or whatever type you want) stored in a container named "PublicFiles" on your storage account in Azure.

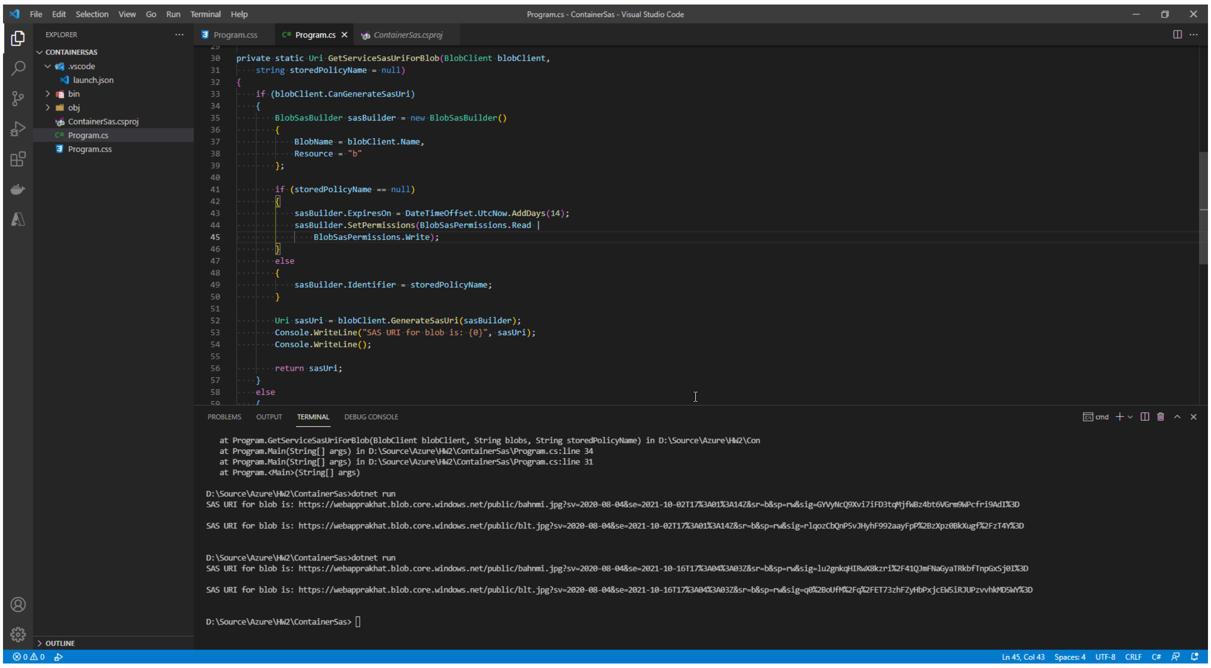
- These files have to be unavailable publicly (have to be private) and only your application has to be able to get access to them.
- Even more, these files have to be accessible for a limited period of time (2 weeks) and your application has to be allowed only to read from this container.

As a report, please, attach code of your application and screenshots of your web page, storage account and/or container containing information about your Azure account (user name) and explanation of the implemented solution.

Tip: think about **Shared access signature (SAS).**

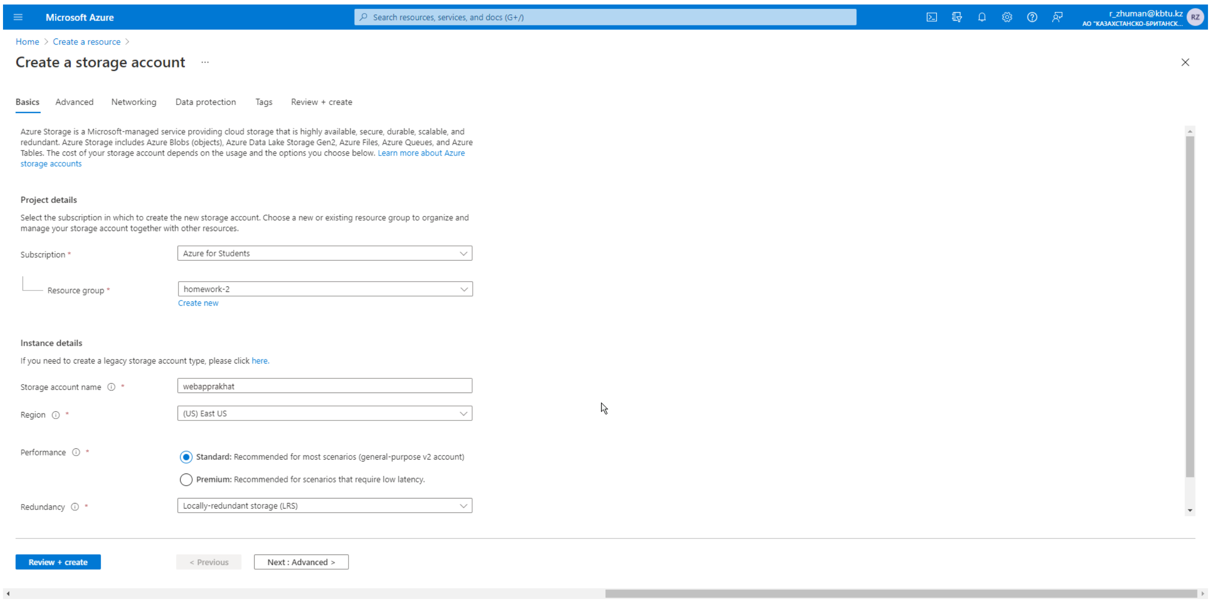
Report

It took me a day to understand the C# language and at the end I got stuck on how to make a page, so I had to start over, but with Javascript, which is already close to me.



```
private static Uri GetServiceSasUriForBlob(BlobClient blobClient,
    string storedPolicyName = null)
{
    if (blobClient.CanGenerateSasUri)
    {
        BlobSasBuilder sasBuilder = new BlobSasBuilder()
        {
            BlobName = blobClient.Name,
            Resource = "b"
        };

        if (storedPolicyName == null)
        {
            sasBuilder.ExpiresOn = DateTimeOffset.UtcNow.AddDays(14);
            sasBuilder.SetPermissions(BlobSasPermissions.Read |
                BlobSasPermissions.Write);
        }
        else
        {
            sasBuilder.Identifier = storedPolicyName;
        }

        Uri sasUri = blobClient.GenerateSasUri(sasBuilder);
        Console.WriteLine("SAS URI for blob is: {0}", sasUri);
        Console.WriteLine();

        return sasUri;
    }
    else
```

```
   at Program.GetServiceSasUriForBlob(BlobClient blobClient, String blobs, String storedPolicyName) in D:\Source\Azure\HW2\Con
   at Program.Main(String[] args) in D:\Source\Azure\HW2\ContainerSas\Program.cs:line 34
   at Program.Main(String[] args) in D:\Source\Azure\HW2\ContainerSas\Program.cs:line 31
   at Program.<Main>(String[] args)

D:\Source\Azure\HW2\ContainerSas>dotnet run
SAS URI for blob is: https://webapprakhat.blob.core.windows.net/public/bahnmi.jpg?sv=2020-08-04&se=2021-10-02T17%3A01%3A14Z&sr=b&sp=rw&sig=GYVyNcQ9Xvi7iFD3tqMjfWBz4bt6VGrm9WPcfri9AdI%3D

SAS URI for blob is: https://webapprakhat.blob.core.windows.net/public/blt.jpg?sv=2020-08-04&se=2021-10-02T17%3A01%3A14Z&sr=b&sp=rw&sig=rlqozCbQnP5vJHyhF992aayFpP%2BzXpz0BkXugf%2FzT4Y%3D


D:\Source\Azure\HW2\ContainerSas>dotnet run
SAS URI for blob is: https://webapprakhat.blob.core.windows.net/public/bahnmi.jpg?sv=2020-08-04&se=2021-10-16T17%3A04%3A03Z&sr=b&sp=rw&sig=lu2gnkqHIRwX8kzri%2F41QJmFNaGyaTRkbfTnpGxSj0I%3D

SAS URI for blob is: https://webapprakhat.blob.core.windows.net/public/blt.jpg?sv=2020-08-04&se=2021-10-16T17%3A04%3A03Z&sr=b&sp=rw&sig=q0%2BoUfM%2Fq%2FET73zhFZyHbPxjcEW5iRJUPzvvhkMD5WY%3D


D:\Source\Azure\HW2\ContainerSas> 
```



Create a storage account

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. Learn more about Azure storage accounts

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription * Azure for Students

Resource group * homework-2

Create new

Instance details

If you need to create a legacy storage account type, please click here.

Storage account name * webapprakhat

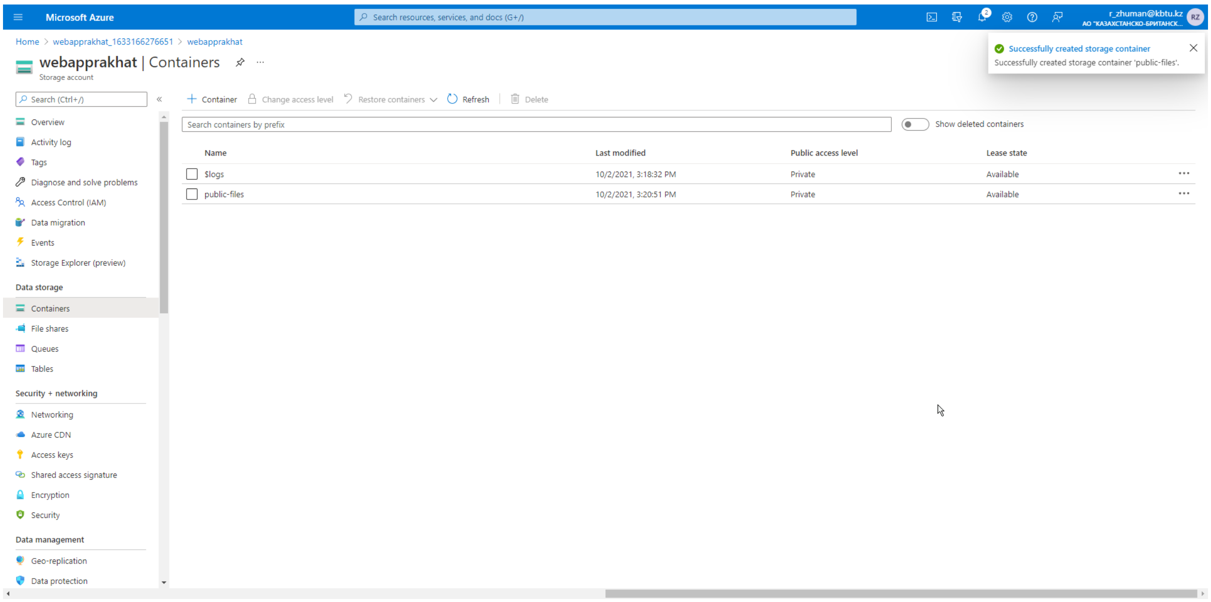
Region * (US) East US

Performance *

● **Standard:** Recommended for most scenarios (general-purpose v2 account)

○ **Premium:** Recommended for scenarios that require low latency.

Redundancy * Locally-redundant storage (LRS)

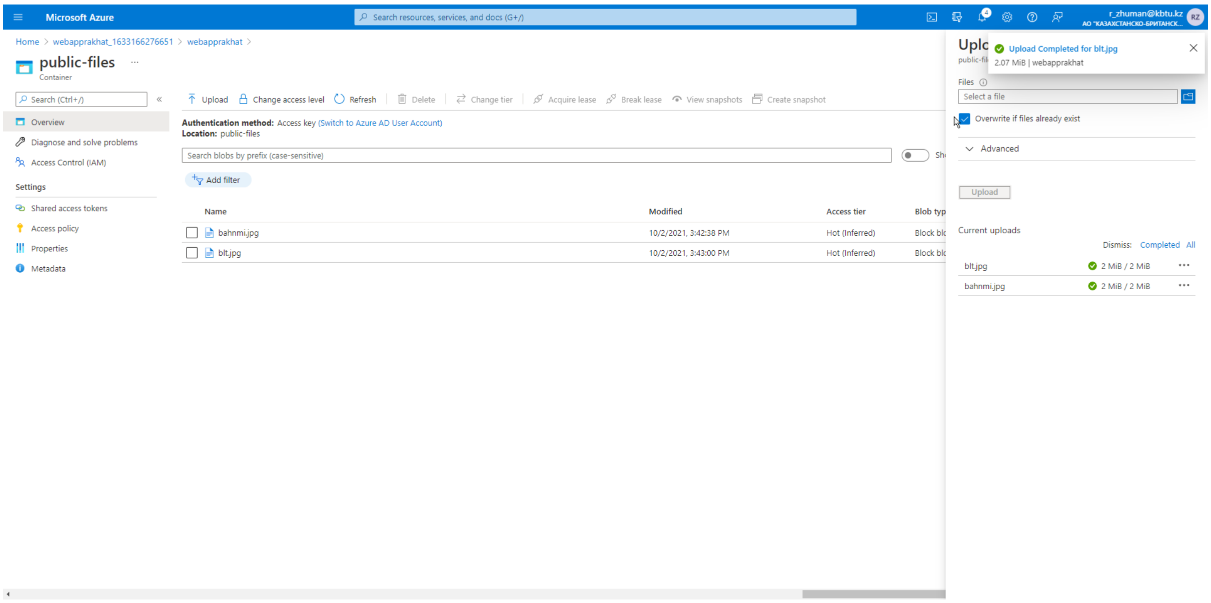


webapprakhat | Containers

Successfully created storage container

Successfully created storage container 'public-files'.

Name	Last modified	Public access level	Lease state
□ $logs	10/2/2021, 3:18:32 PM	Private	Available
□ public-files	10/2/2021, 3:20:51 PM	Private	Available



Microsoft Azure

Home > webapprakhat_1633166276651 > webapprakhat >

public-files

Container

Upload | Change access level | Refresh | Delete | Change tier | Acquire lease | Break lease | View snapshots | Create snapshot

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Shared access tokens

Access policy

Properties

Metadata

Authentication method: Access key (Switch to Azure AD User Account)
Location: public-files

Search blobs by prefix (case-sensitive)

	Name	Modified	Access tier	Blob typ
□	bahnmi.jpg	10/2/2021, 3:42:38 PM	Hot (Inferred)	Block blo
□	blt.jpg	10/2/2021, 3:43:00 PM	Hot (Inferred)	Block blo

Upload Completed for blt.jpg
2.07 MiB | webapprakhat

Files

Select a file

☑ Overwrite if files already exist

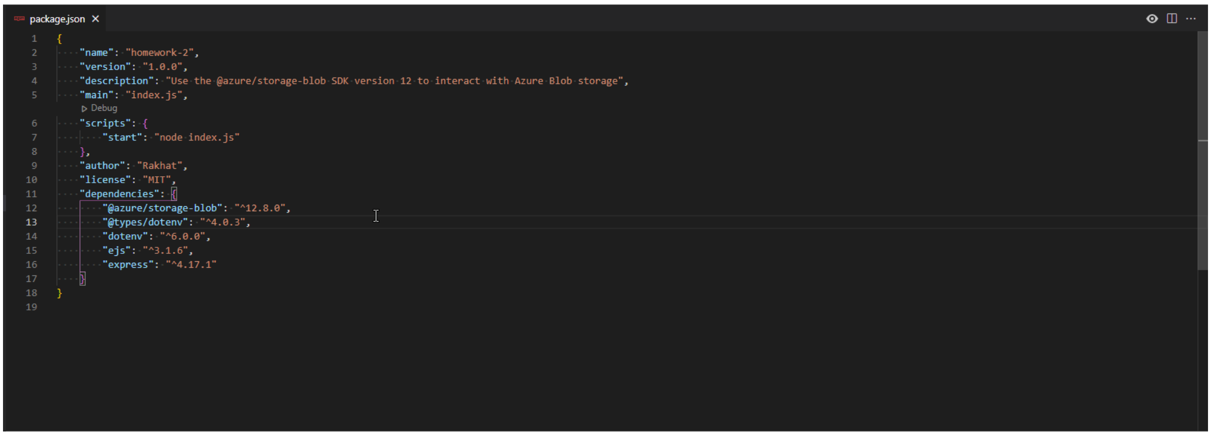
Advanced

Upload

Current uploads

Dismiss: Completed All

blt.jpg	✓ 2 MiB / 2 MiB
bahnmi.jpg	✓ 2 MiB / 2 MiB

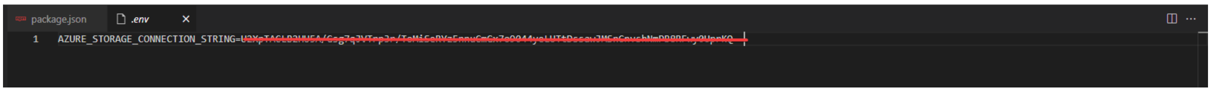


package.json

```
{
    "name": "homework-2",
    "version": "1.0.0",
    "description": "Use the @azure/storage-blob SDK version 12 to interact with Azure Blob storage",
    "main": "index.js",
    ▷ Debug
    "scripts": {
        "start": "node index.js"
    },
    "author": "Rakhat",
    "license": "MIT",
    "dependencies": {
        "@azure/storage-blob": "^12.8.0",
        "@types/dotenv": "^4.0.3",
        "dotenv": "^6.0.0",
        "ejs": "^3.1.6",
        "express": "^4.17.1"
    }
}
```

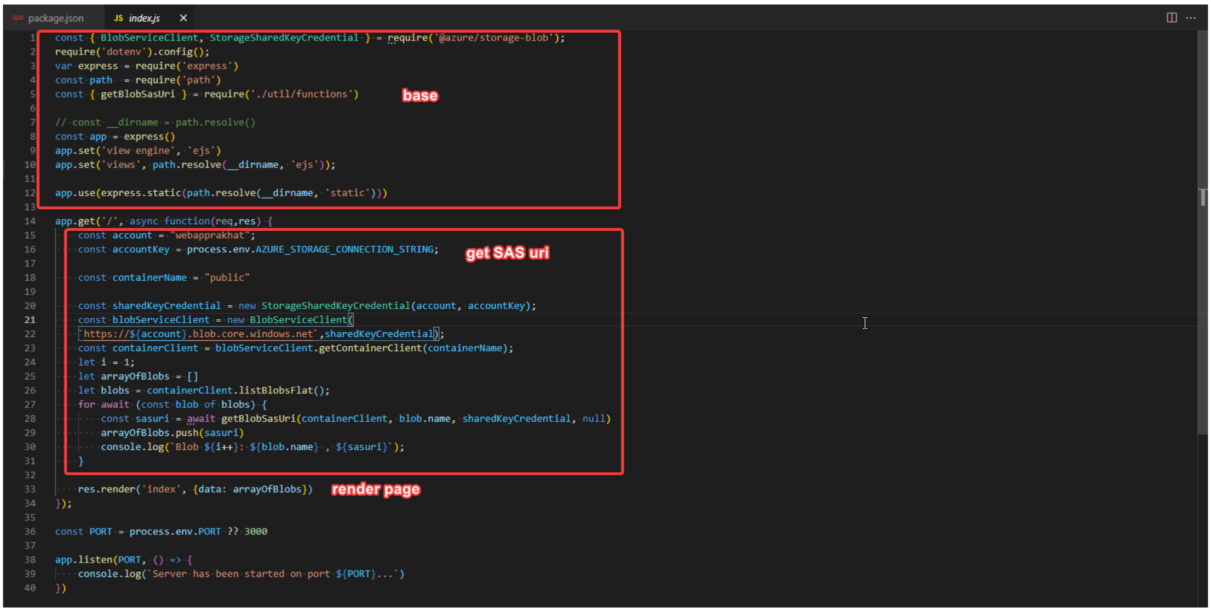
>> npm install

Create .env file for connection string



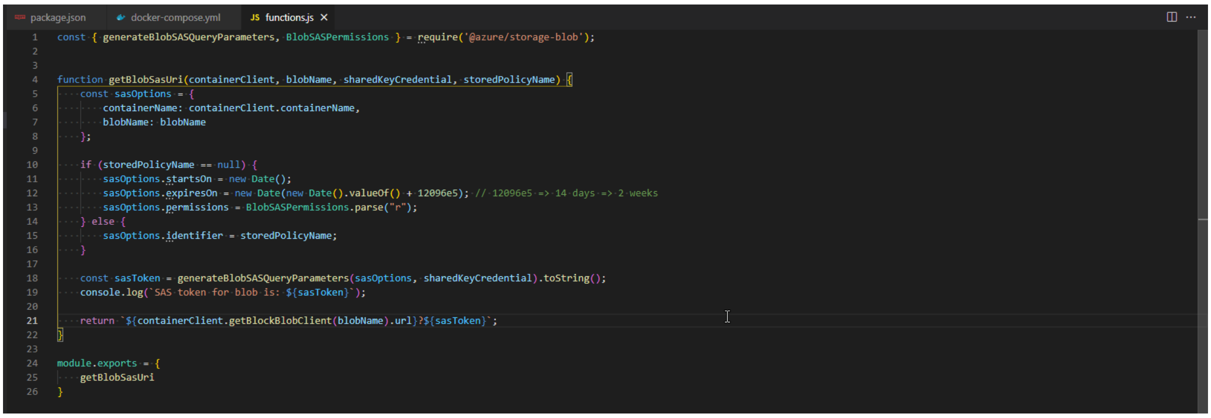
package.json | .env

```
AZURE_STORAGE_CONNECTION_STRING=[redacted]
```



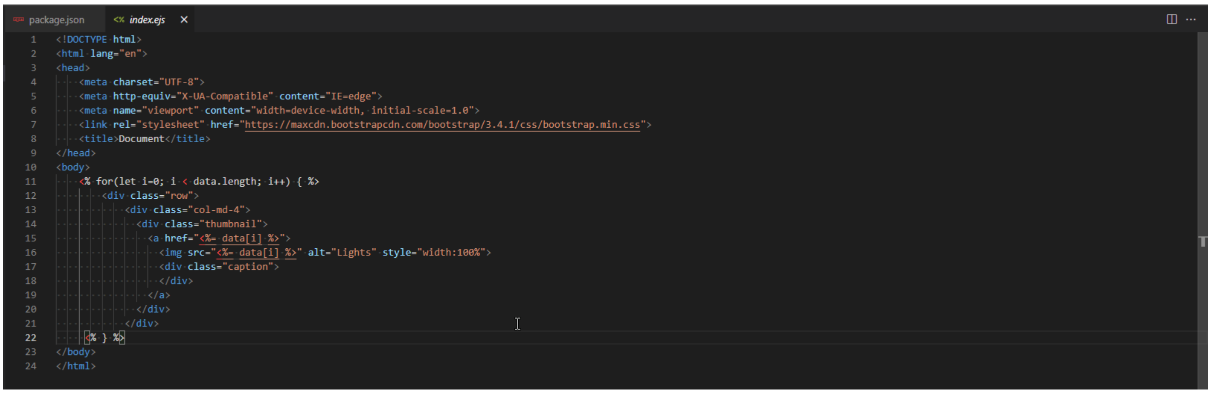
```javascript
const { BlobServiceClient, StorageSharedKeyCredential } = require('@azure/storage-blob');
require('dotenv').config();
var express = require('express')
const path  = require('path')
const { getBlobSasUri } = require('./util/functions')

// const __dirname = path.resolve()
const app = express()
app.set('view engine', 'ejs')
app.set('views', path.resolve(__dirname, 'ejs'));

app.use(express.static(path.resolve(__dirname, 'static')))

app.get('/', async function(req,res) {
    const account = "webapprakhat";
    const accountKey = process.env.AZURE_STORAGE_CONNECTION_STRING;

    const containerName = "public"

    const sharedKeyCredential = new StorageSharedKeyCredential(account, accountKey);
    const blobServiceClient = new BlobServiceClient(
    `https://${account}.blob.core.windows.net`,sharedKeyCredential);
    const containerClient = blobServiceClient.getContainerClient(containerName);
    let i = 1;
    let arrayOfBlobs = []
    let blobs = containerClient.listBlobsFlat();
    for await (const blob of blobs) {
        const sasuri = await getBlobSasUri(containerClient, blob.name, sharedKeyCredential, null)
        arrayOfBlobs.push(sasuri)
        console.log(`Blob ${i++}: ${blob.name} , ${sasuri}`);
    }

    res.render('index', {data: arrayOfBlobs})
});

const PORT = process.env.PORT ?? 3000

app.listen(PORT, () => {
    console.log(`Server has been started on port ${PORT}...`)
})
```

SAS generator function (2 weeks and read permission)



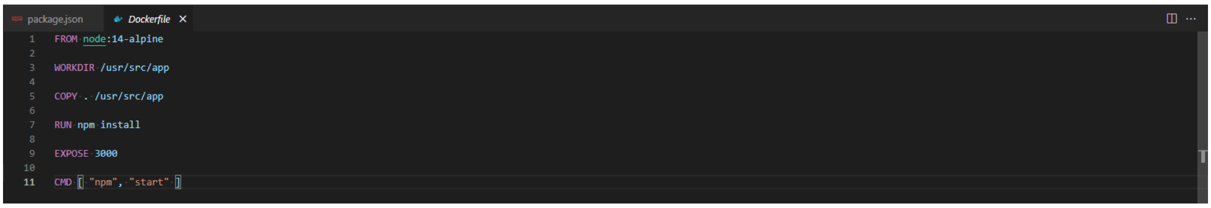
```javascript
const { generateBlobSASQueryParameters, BlobSASPermissions } = require('@azure/storage-blob');


function getBlobSasUri(containerClient, blobName, sharedKeyCredential, storedPolicyName) {
    const sasOptions = {
        containerName: containerClient.containerName,
        blobName: blobName
    };

    if (storedPolicyName == null) {
        sasOptions.startsOn = new Date();
        sasOptions.expiresOn = new Date(new Date().valueOf() + 12096e5); // 12096e5 => 14 days => 2 weeks
        sasOptions.permissions = BlobSASPermissions.parse("r");
    } else {
        sasOptions.identifier = storedPolicyName;
    }

    const sasToken = generateBlobSASQueryParameters(sasOptions, sharedKeyCredential).toString();
    console.log(`SAS token for blob is: ${sasToken}`);

    return `${containerClient.getBlockBlobClient(blobName).url}?${sasToken}`;
}

module.exports = {
    getBlobSasUri
}
```

Easiest one



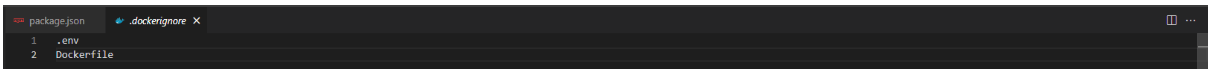
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <title>Document</title>
</head>
<body>
    <% for(let i=0; i < data.length; i++) { %>
        <div class="row">
            <div class="col-md-4">
              <div class="thumbnail">
                <a href="<%= data[i] %>">
                  <img src="<%= data[i] %>" alt="Lights" style="width:100%">
                  <div class="caption">
                  </div>
                </a>
              </div>
            </div>
     <% } %>
</body>
</html>
```

Dockerfile



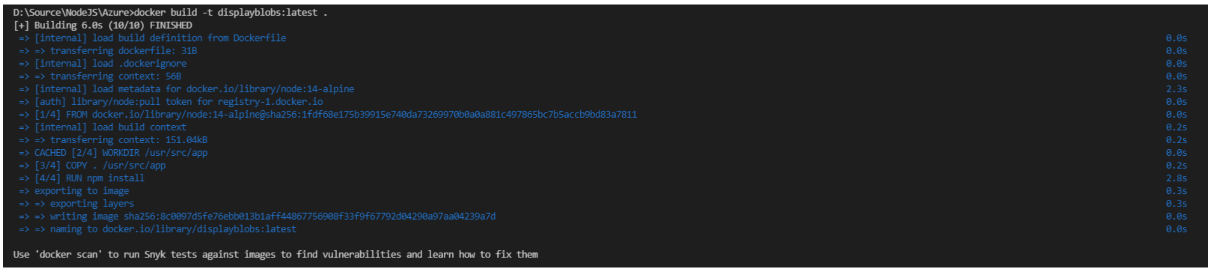
```dockerfile
FROM node:14-alpine

WORKDIR /usr/src/app

COPY . /usr/src/app

RUN npm install

EXPOSE 3000

CMD [ "npm", "start" ]
```

I am going to publish in Docker Hub, so ignore .env file

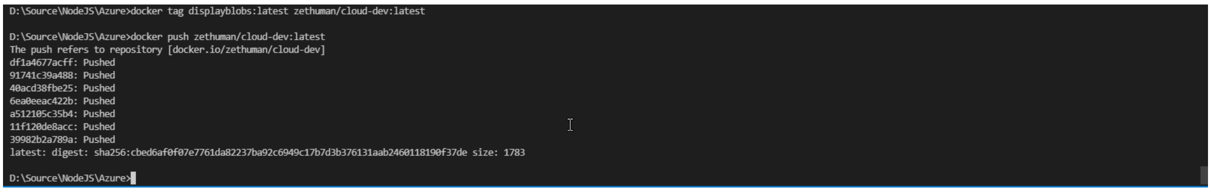


```
.env
Dockerfile
```

Build & push

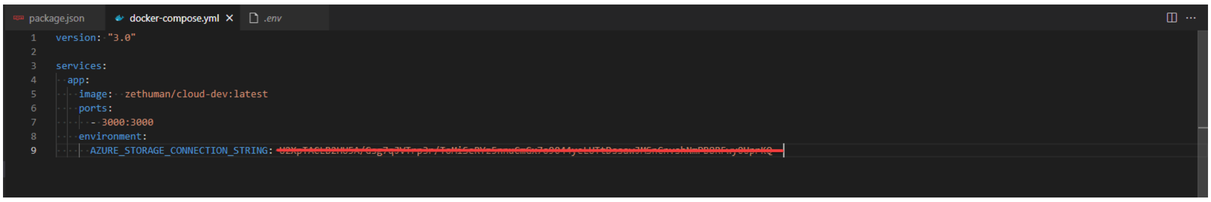


```
D:\Source\NodeJS\Azure>docker build -t displayblobs:latest .
[+] Building 6.0s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                                                         0.0s
 => => transferring dockerfile: 31B                                                                          0.0s
 => [internal] load .dockerignore                                                                            0.0s
 => => transferring context: 56B                                                                             0.0s
 => [internal] load metadata for docker.io/library/node:14-alpine                                            2.3s
 => [auth] library/node:pull token for registry-1.docker.io                                                  0.0s
 => [1/4] FROM docker.io/library/node:14-alpine@sha256:1fdf68e175b39915e740da73269970b0a0a881c497865bc7b5accb9bd83a7811  0.0s
 => [internal] load build context                                                                            0.2s
 => => transferring context: 151.04kB                                                                        0.2s
 => CACHED [2/4] WORKDIR /usr/src/app                                                                        0.0s
 => [3/4] COPY . /usr/src/app                                                                                0.2s
 => [4/4] RUN npm install                                                                                    2.8s
 => exporting to image                                                                                       0.3s
 => => exporting layers                                                                                      0.3s
 => => writing image sha256:8c0097d5fe76ebb013b1aff44867756908f33f9f67792d04290a97aa04239a7d                 0.0s
 => => naming to docker.io/library/displayblobs:latest                                                       0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```



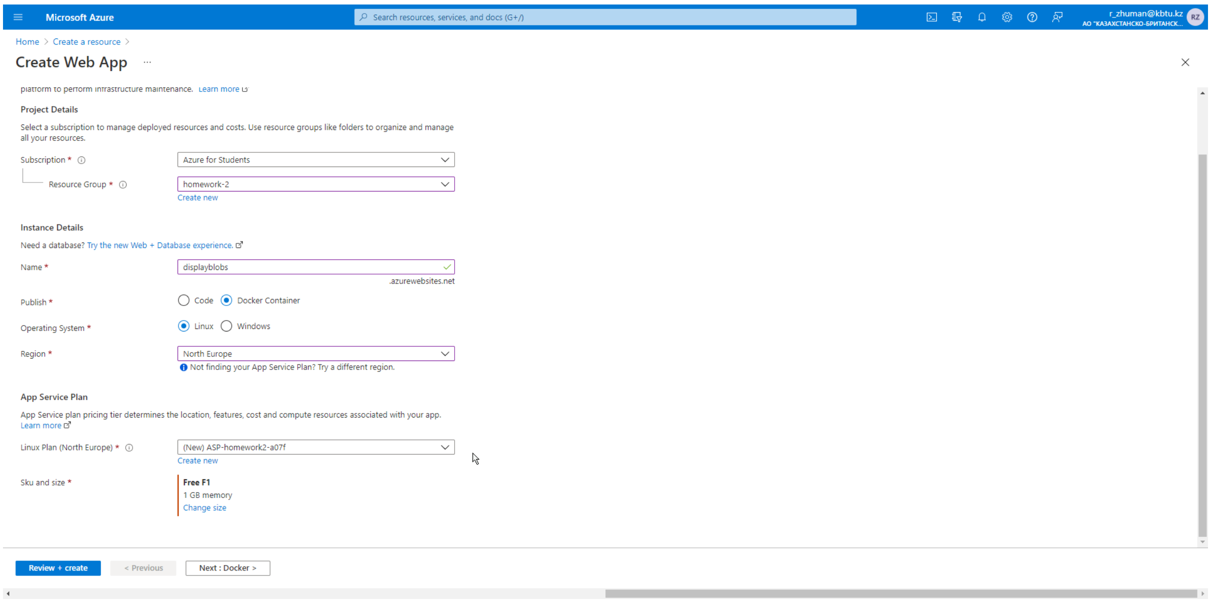
```
D:\Source\NodeJS\Azure>docker tag displayblobs:latest zethuman/cloud-dev:latest

D:\Source\NodeJS\Azure>docker push zethuman/cloud-dev:latest
The push refers to repository [docker.io/zethuman/cloud-dev]
df1a4677acff: Pushed
91741c39a488: Pushed
40acd38fbe25: Pushed
6ea0eeac422b: Pushed
a512105c35b4: Pushed
11f120de8acc: Pushed
39982b2a789a: Pushed
latest: digest: sha256:cbed6af0f07e7761da82237ba92c6949c17b7d3b376131aab2460118190f37de size: 1783

D:\Source\NodeJS\Azure>
```

docker-compose



```yaml
version: "3.0"

services:
  app:
    image: zethuman/cloud-dev:latest
    ports:
      - 3000:3000
    environment:
      AZURE_STORAGE_CONNECTION_STRING: [redacted]
```

Create Web App


Microsoft Azure
Search resources, services, and docs (G+/)
r_zhuman@kbtu.kz
АО "КАЗАХСТАНСКО-БРИТАНСК...
RZ
Home > Create a resource >

Create Web App ···

platform to perform infrastructure maintenance. Learn more

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Azure for Students

Resource Group * : homework-2

Create new

Instance Details

Need a database? Try the new Web + Database experience.

Name * : displayblobs .azurewebsites.net

Publish * : ○ Code ● Docker Container

Operating System * : ● Linux ○ Windows

Region * : North Europe

Not finding your App Service Plan? Try a different region.

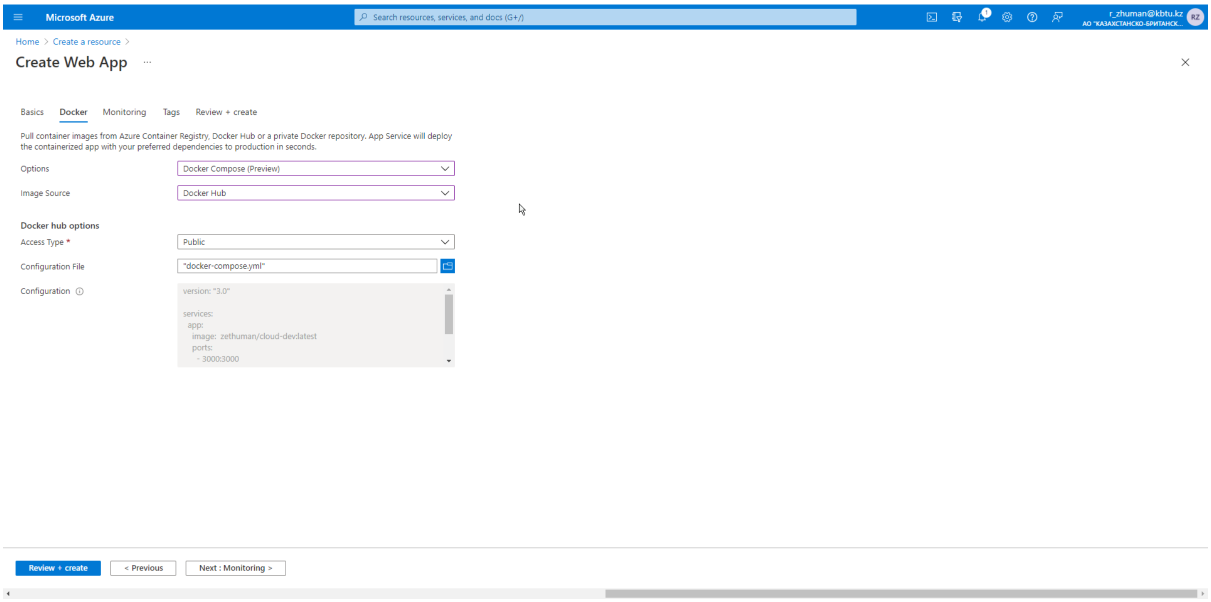
App Service Plan

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. Learn more

Linux Plan (North Europe) * : (New) ASP-homework2-a07f

Create new

Sku and size * : **Free F1**
1 GB memory
Change size



Microsoft Azure
Search resources, services, and docs (G+/)
r_zhuman@kbtu.kz
АО "КАЗАХСТАНСКО-БРИТАНСК...
RZ
Home > Create a resource >

Create Web App

Basics
Docker
Monitoring
Tags
Review + create

Pull container images from Azure Container Registry, Docker Hub or a private Docker repository. App Service will deploy the containerized app with your preferred dependencies to production in seconds.

Options: Docker Compose (Preview)

Image Source: Docker Hub

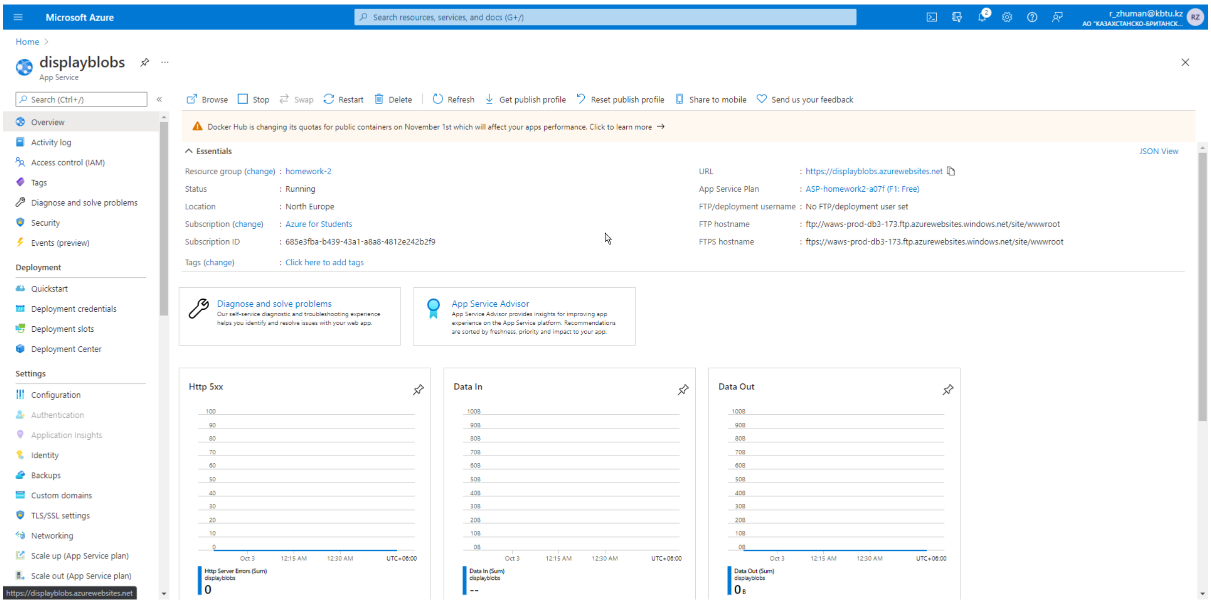
Docker hub options

Access Type *: Public

Configuration File: "docker-compose.yml"

Configuration:

```
version: "3.0"

services:
  app:
    image:  zethuman/cloud-dev:latest
    ports:
     - 3000:3000
```



Microsoft Azure
Search resources, services, and docs (G+/)
r_zhuman@kbtu.kz
АО "КАЗАХСТАНСКО-БРИТАНСК...
RZ
Home >

displayblobs
App Service

Docker Hub is changing its quotas for public containers on November 1st which will affect your apps performance. Click to learn more →

Essentials

JSON View

Resource group (change) : homework-2

Status : Running

Location : North Europe

Subscription (change) : Azure for Students

Subscription ID : 685e3fba-b439-43a1-a8a8-4812e242b2f9

Tags (change) : Click here to add tags

URL : https://displayblobs.azurewebsites.net

App Service Plan : ASP-homework2-a07f (F1: Free)

FTP/deployment username : No FTP/deployment user set

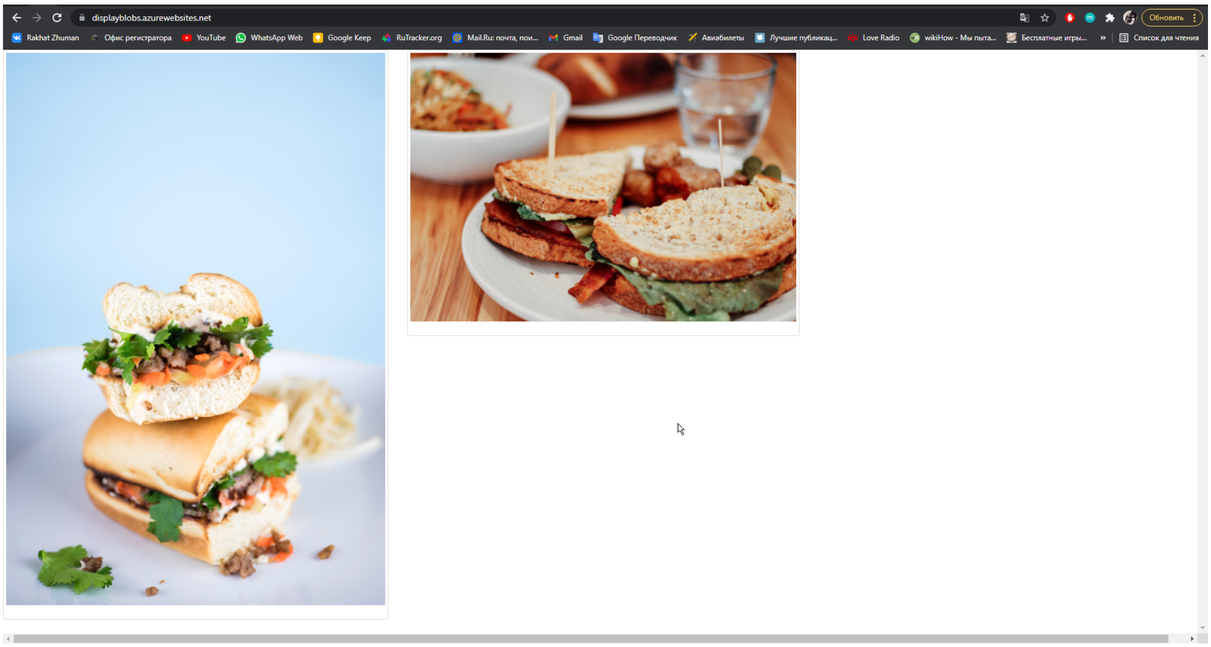
FTP hostname : ftp://waws-prod-db3-173.ftp.azurewebsites.windows.net/site/wwwroot

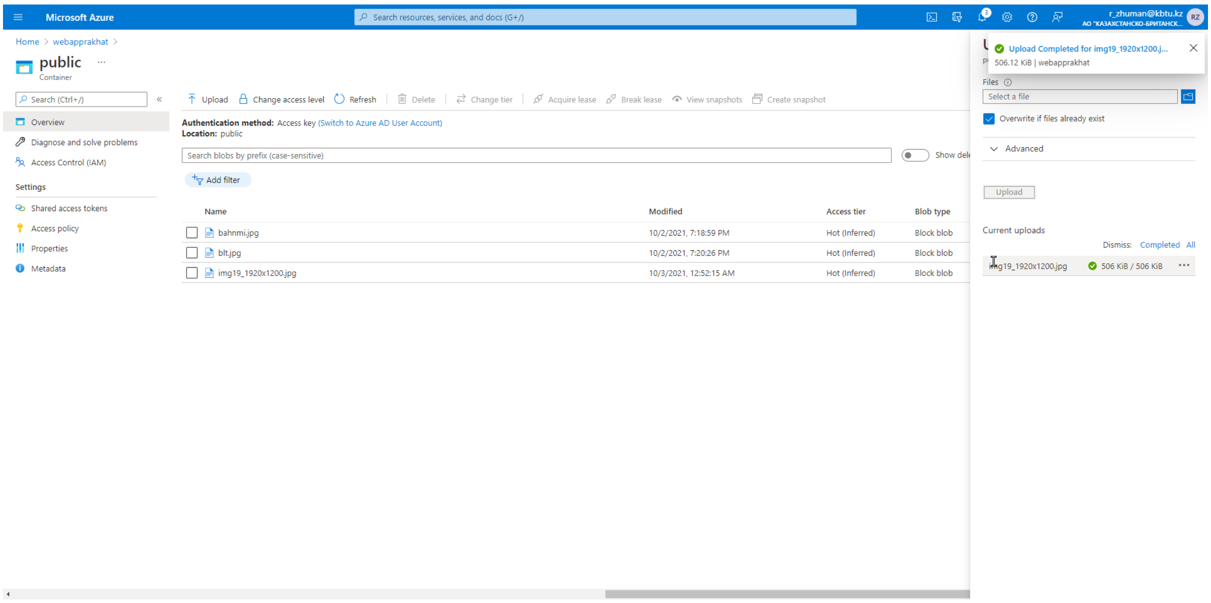
FTPS hostname : ftps://waws-prod-db3-173.ftp.azurewebsites.windows.net/site/wwwroot

Diagnose and solve problems
Our self-service diagnostic and troubleshooting experience helps you identify and resolve issues with your web app.

App Service Advisor
App Service Advisor provides insights for improving app experience on the App Service platform. Recommendations are sorted by freshness, priority and impact to your app.

And finally…




Microsoft Azure
Search resources, services, and docs (G+/)
r_zhuman@kbtu.kz
АО "КАЗАХСТАНСКО-БРИТАНСК...

Home > webapprakhat >

public ···
Container

Search (Ctrl+/)
Overview
Diagnose and solve problems
Access Control (IAM)
Settings
Shared access tokens
Access policy
Properties
Metadata

Upload | Change access level | Refresh | Delete | Change tier | Acquire lease | Break lease | View snapshots | Create snapshot

Authentication method: Access key (Switch to Azure AD User Account)
Location: public

Search blobs by prefix (case-sensitive)

Show dele

Add filter

	Name	Modified	Access tier	Blob type
□	bahnmi.jpg	10/2/2021, 7:18:59 PM	Hot (Inferred)	Block blob
□	blt.jpg	10/2/2021, 7:20:26 PM	Hot (Inferred)	Block blob
□	img19_1920x1200.jpg	10/3/2021, 12:52:15 AM	Hot (Inferred)	Block blob

Upload Completed for img19_1920x1200.j...
506.12 KiB | webapprakhat

Files
Select a file

☑ Overwrite if files already exist

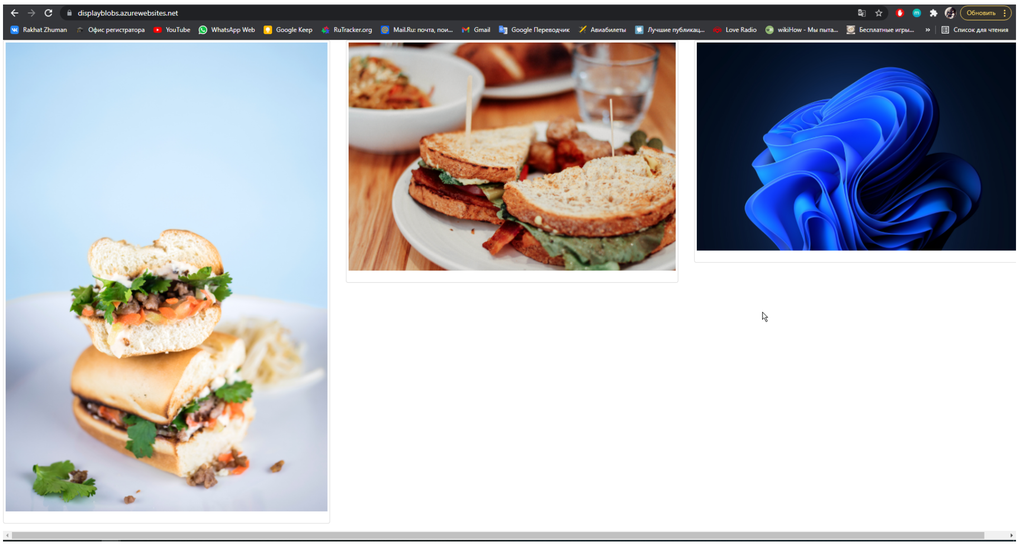
Advanced

Upload

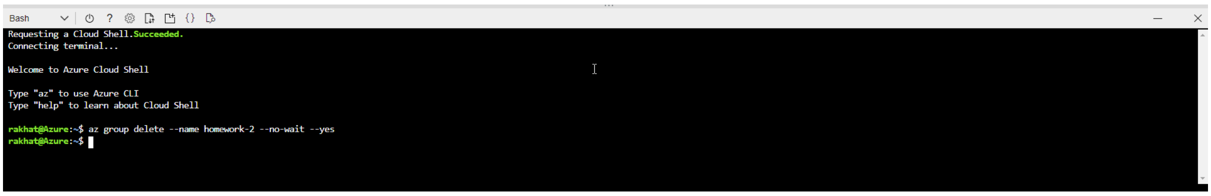
Current uploads

Dismiss: Completed All

img19_1920x1200.jpg ✓ 506 KiB / 506 KiB ···



Cleaning up workspace



```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

rakhat@Azure:~$ az group delete --name homework-2 --no-wait --yes
rakhat@Azure:~$ 
```