Cloud Development, 2021 fall.
Homework 3

Report

Storages

made by Zhuman Rakhat

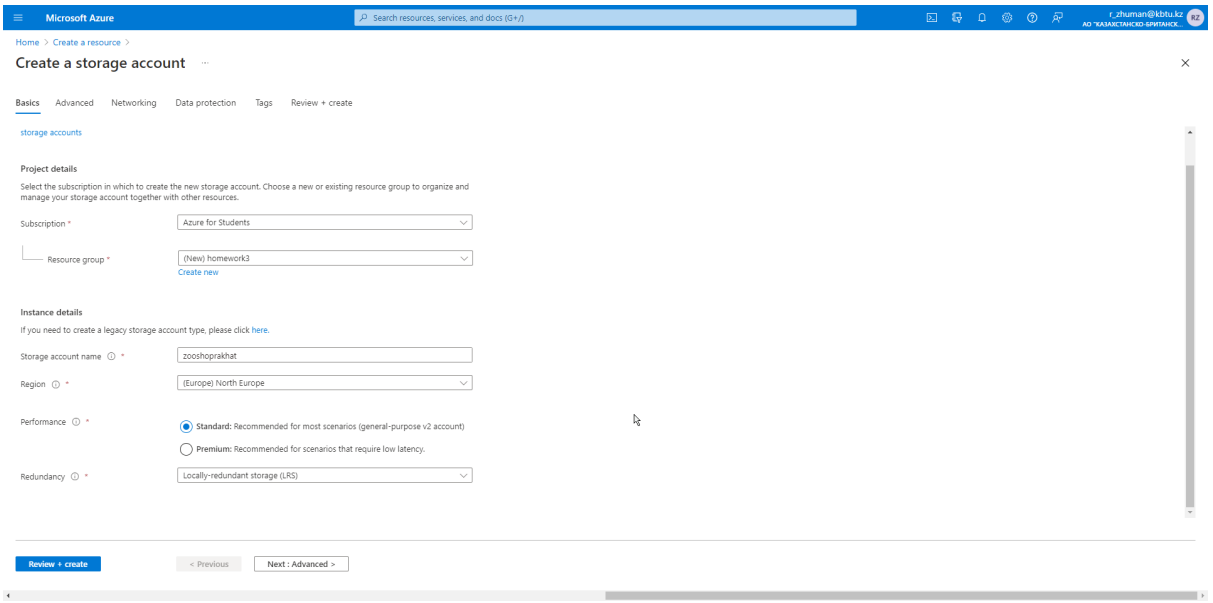
Almaty 2021

Task

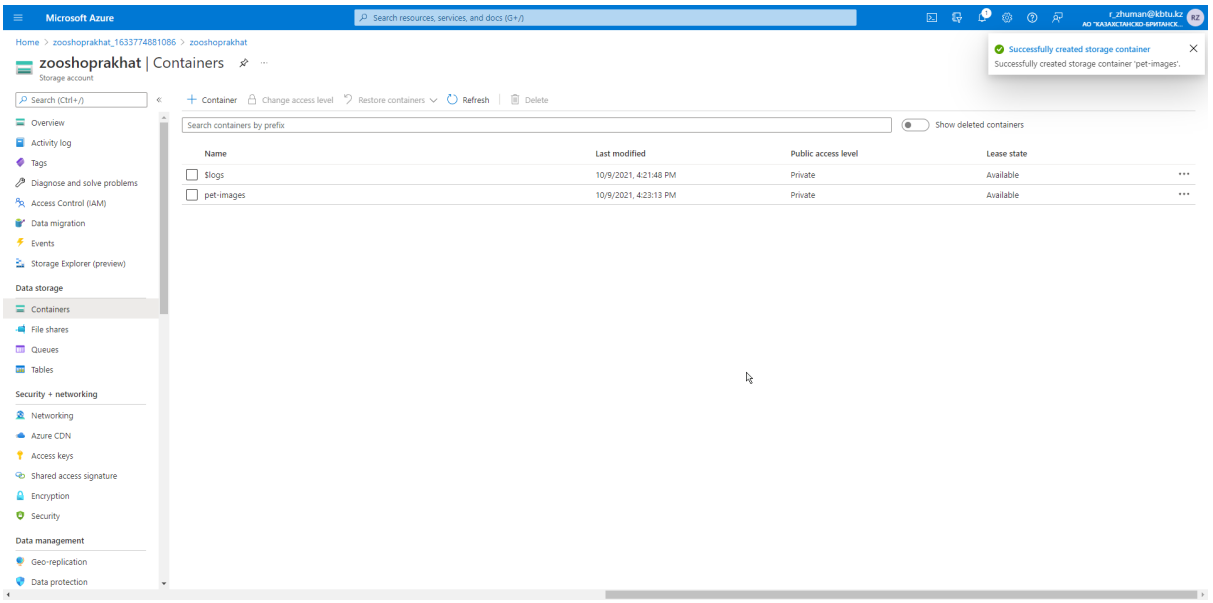
You need to create a page for Zoo-shop, displaying information about all pets in the shop (if you will add a pet photo it will be even better). It has to let to edit, remove and add information. Please, make sure, that this page has to be published as Azure Web App, which is using Azure Cosmos DB for data storing and Azure Blob Storage as an images storage. As a report, please attach screenshots of the services used, as well as the code of your solution.

Report

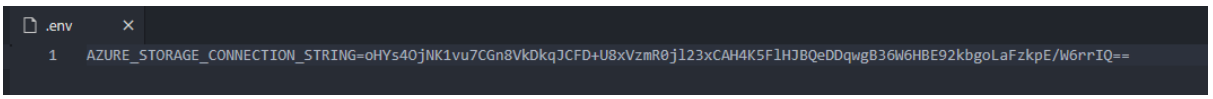
Create Storage Account



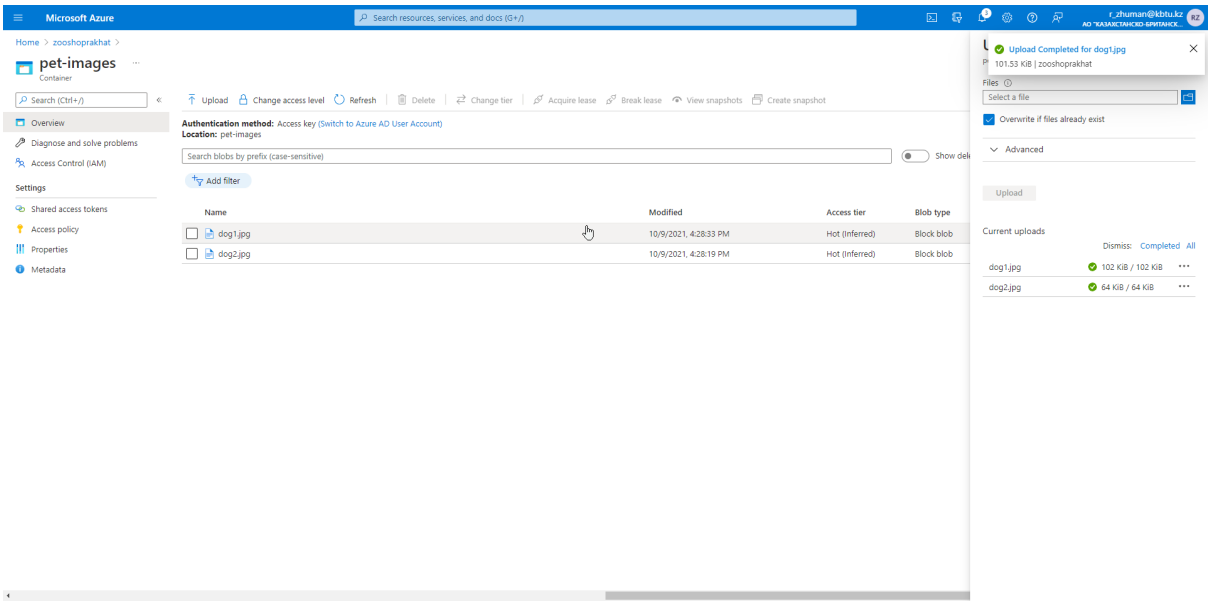
Create Container pet-images



Store key to .env file



Upload images

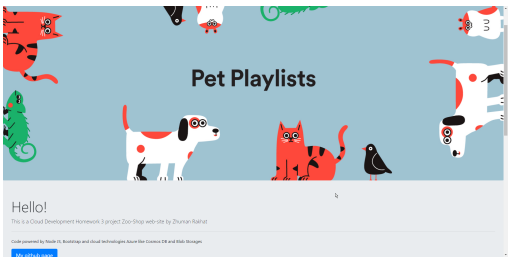


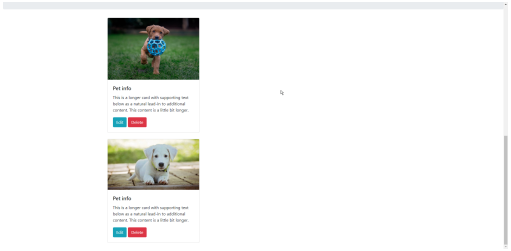
Create Node JS app and design site with bootstrap

```
const { BlobServiceClient, StorageSharedKeyCredential } = require('@azure/storage-blob');
require('dotenv').config();
var express = require('express')
const path  = require('path')
const bodyParser = require('body-parser');

// 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, 'ejs/static')))
app.use(bodyParser.urlencoded({ extended: true }));

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

    const containerName = "pet-images"

    const sharedKeyCredential = new StorageSharedKeyCredential(account, accountKey);
    const blobServiceClient = new BlobServiceClient(
    `https://${account}.blob.core.windows.net`,sharedKeyCredential);
    const containerClient = blobServiceClient.getContainerClient(containerName);
    let i = 1;
    let arrayOfBlobs = []
    let blobs = containerClient.listBlobsFlat();
    for await (const blob of blobs) {
        arrayOfBlobs.push(`https://zooshoprakhat.blob.core.windows.net/pet-images/${blob.name}`)
    }
    console.log(arrayOfBlobs)
    res.render('index', {data: arrayOfBlobs})
});
```

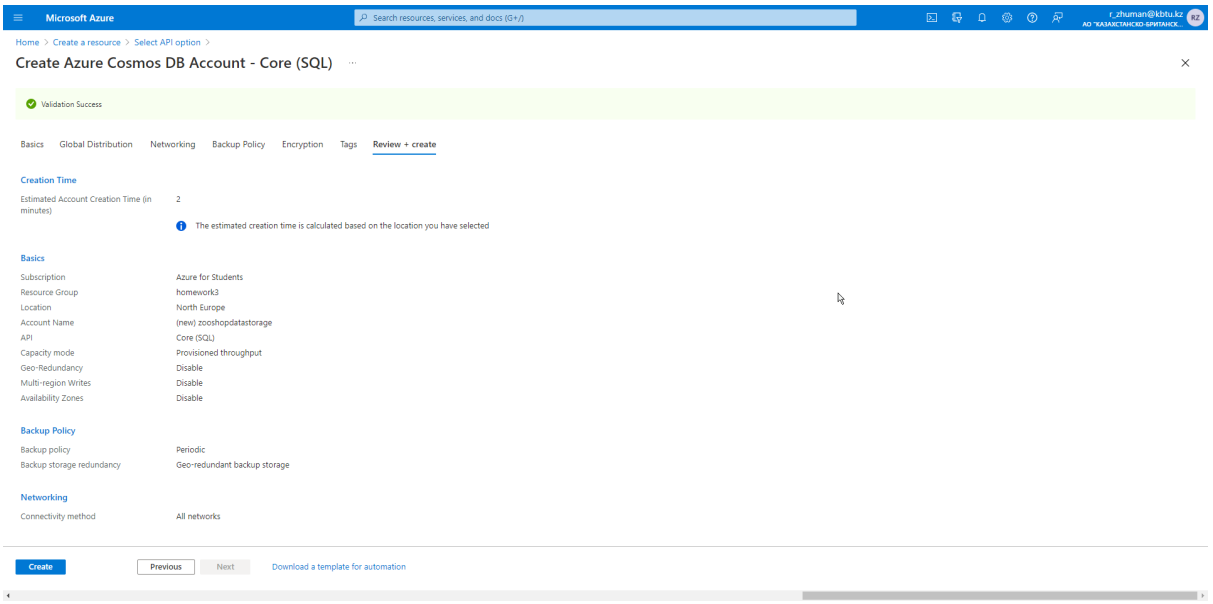
```
<form method="POST" action="deleted" name="delete">
  <% for(let i=0; i < data.length; i++) { %>
  <div class="container">
    <div class="row row-cols-1 row-cols-md-3">
      <div class="col mb-4">
        <div class="card">
          <img src="<%= data[i] %>" class="card-img-top" alt="...">
          <div class="card-body">
            <input hidden name="id" value="12345"/>
            <h5 class="card-title" name="name">Pet info</h5>
            <p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>
            <a href="/edit" class="btn btn-info">Edit</a>
            <button type="SUBMIT" class="btn btn-danger">Delete</button>
          </div>
        </div>
      </div>
    </div>
  </div>
  <% } %>
</form>
```

And result

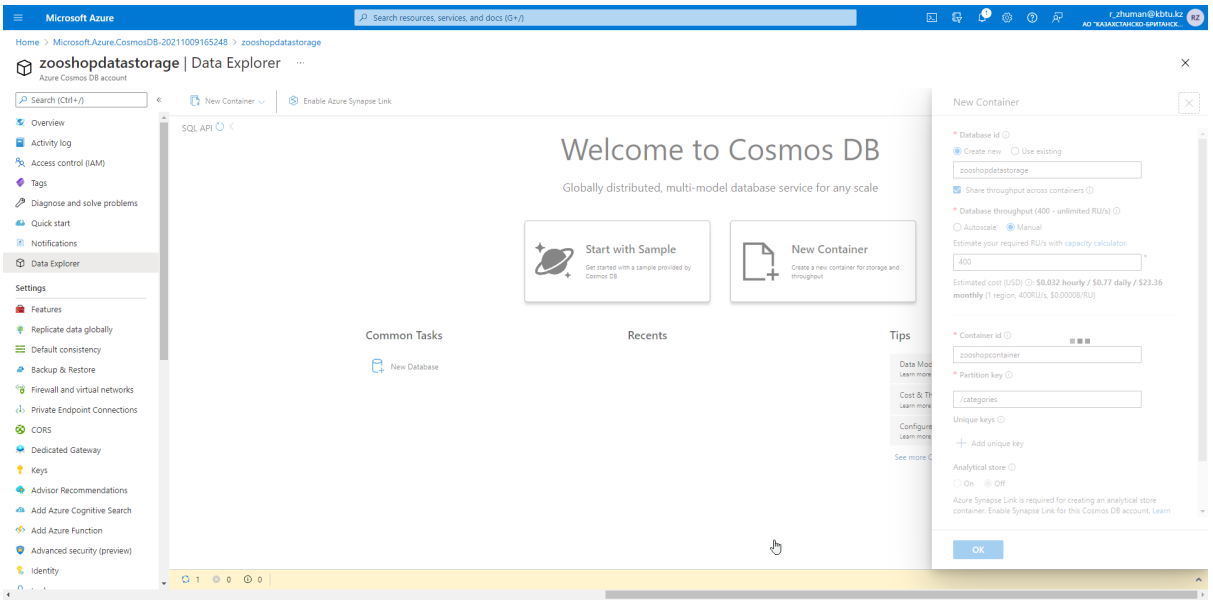




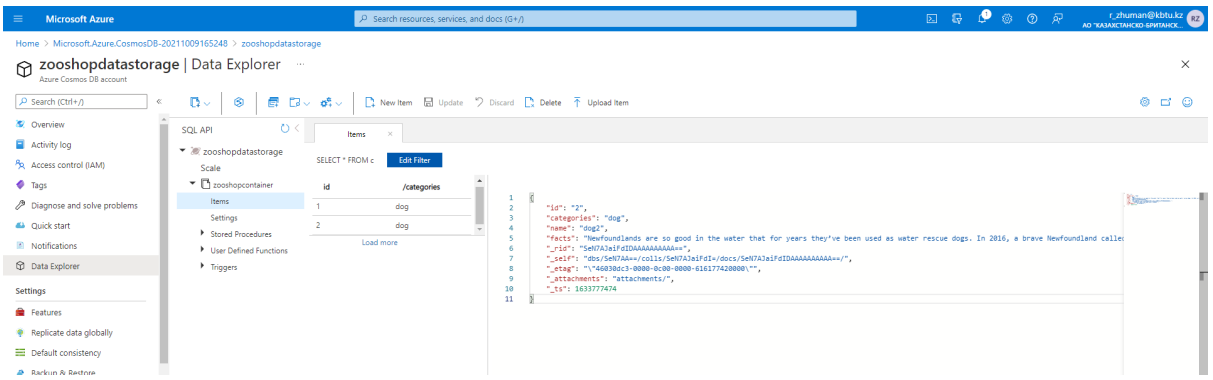
Create Cosmos DB Account



Create a new container



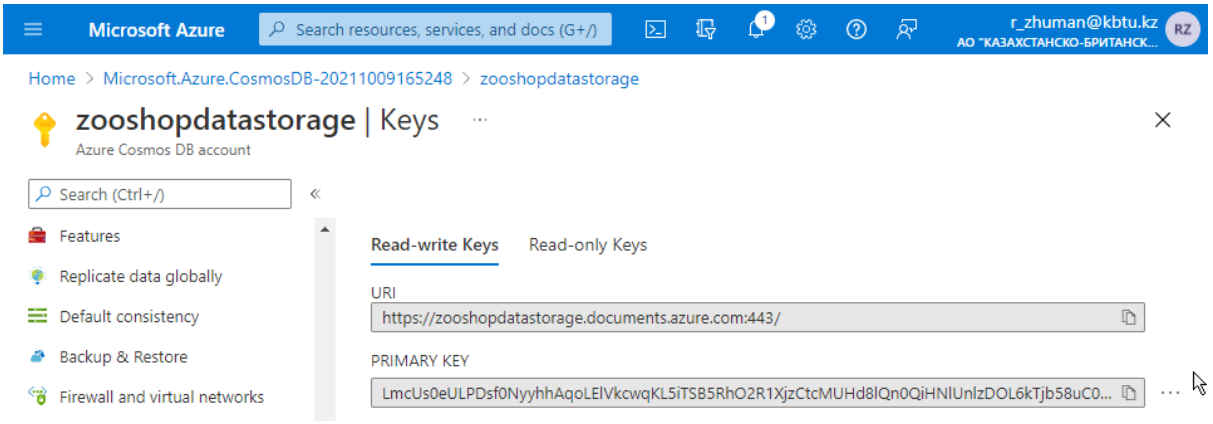
Create a new items

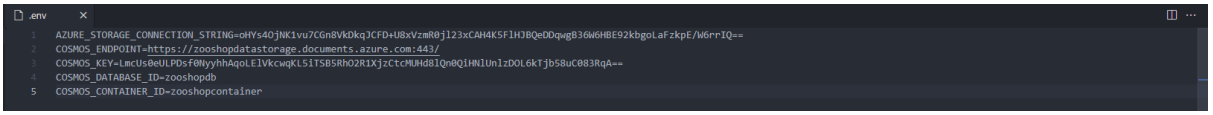


>> npm i @azure/cosmos

```
D:\Source\NodeJS\Azure\HW3>npm i @azure/cosmos
npm WARN homework-2@1.0.0 No repository field.

+ @azure/cosmos@3.14.1
added 20 packages from 8 contributors and audited 121 packages in 10.837s
found 0 vulnerabilities
```

Get credos from project





Long development period

/Add

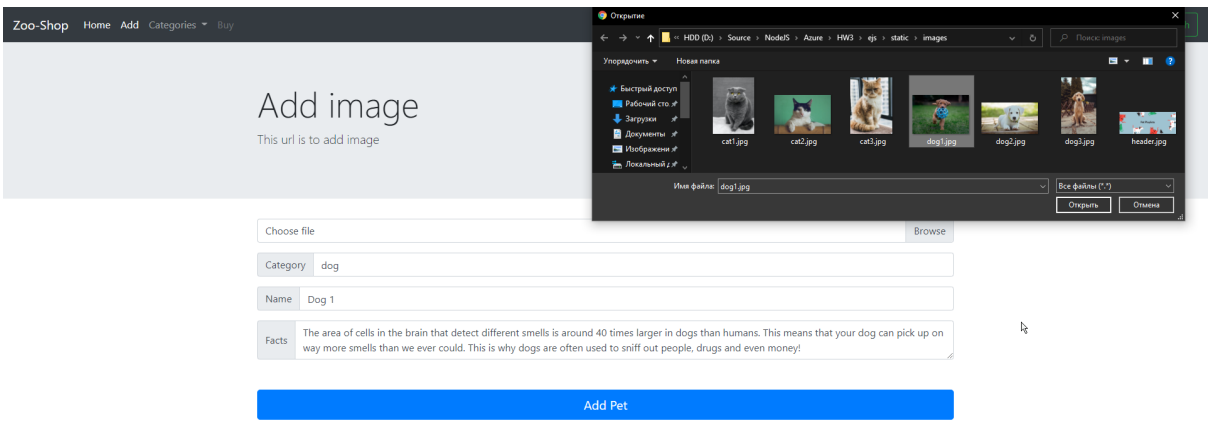
```
app.get('/add', async function(req,res) {
    res.render('add')
});

app.post('/add', singleFileUpload.single('img'), async function(req,res, next) {
    const image = req.file;
    const name = req.body.name;
    const category = req.body.category;
    const facts = req.body.facts;

    try{
        await uploadImage(image)
        await createPet(category, name, image.originalname, facts)
    } catch (err){
        console.log("Error occured when uploading: ", err)
        res.status(400).send("Error occured when uploading: ", err)
    }

    res.render('add')
});
```

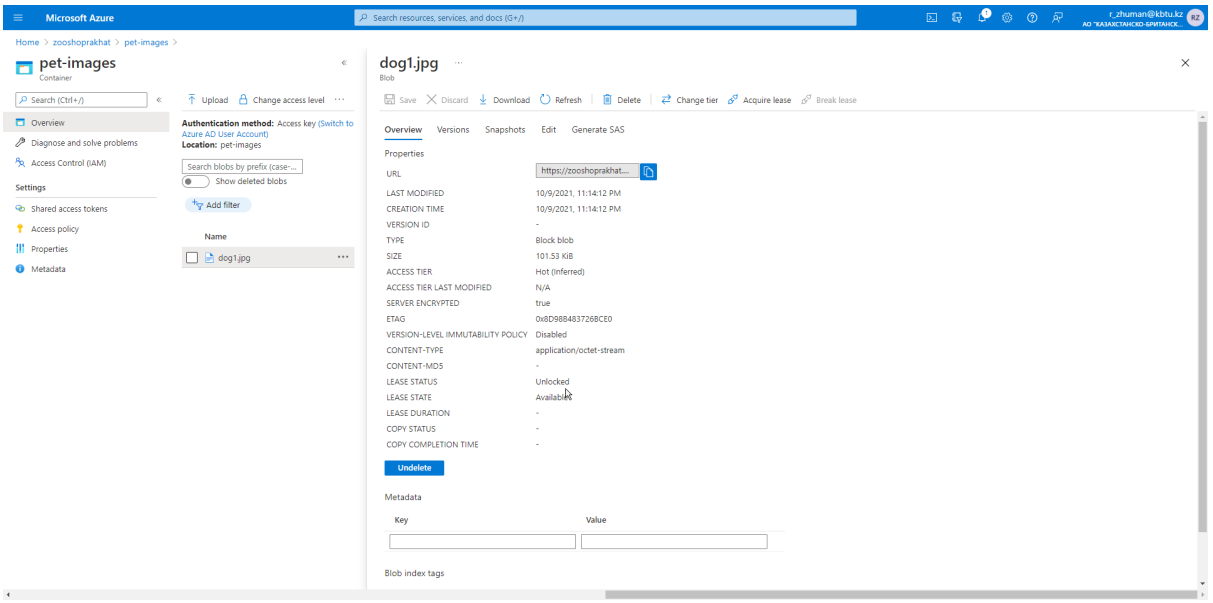
Blob storage create

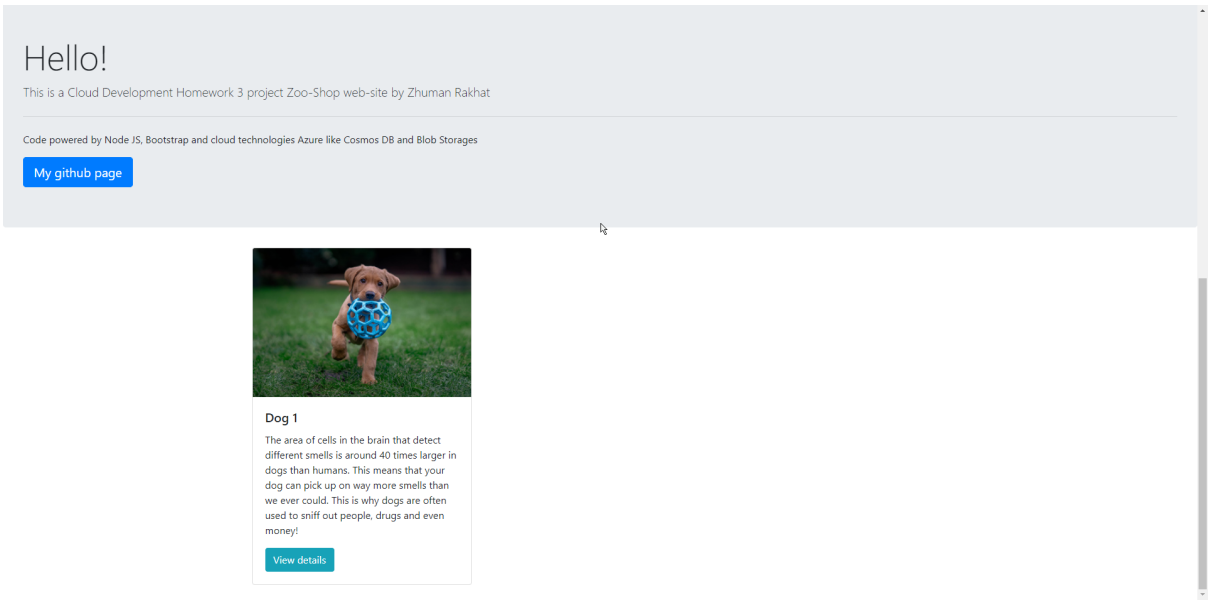
```
async function uploadImage(image){
    const containerClient = await getContainer()

    const blobName = image.originalname;
    const stream = getStream(image.buffer);
    const streamLength = image.size;
    const contentType = image.mimetype;
    const blockBlobClient = containerClient.getBlockBlobClient(blobName);
    blockBlobClient.uploadStream(stream,streamLength)
    console.log(`Image was uploaded`)
}
```

Cosmos create

```
async function createPet(category, name, image, facts){
    const pet = {
        categories: category,
        name: name,
        image: image,
        facts: facts
    }
    return await client.database(databaseId).container(containerId).items.create(pet);
}
```







/Edit

```
app.post('/edit', async function(req,res) {
    console.log('Edit ',req.body)
    const id = req.body.id
    const name = req.body.name
    const facts = req.body.facts
    const category = req.body.category
    const imagename = req.body.imagename
    const method = req.body.method

    if(method==='edit'){
        await updatePet(id, name, facts, category, imagename)
        res.render('edited')
    } else if (method === 'delete'){
        const deleted = await deletePet(id)
        await deleteImage(imagename)
        console.log(deleted)
        res.render('deleted')
    } else{
        res.status(400).render('index')
```

```
async function updatePet(id, name, facts, category, imagename){
    const doc = {}
    doc.id = id
    doc.name = name
    doc.facts = facts
    doc.category = category
    doc.imagename = imagename
    const { result: results } = await client.database(databaseId).container(containerId).item(id).replace(doc)
    return results
}
```

Category dog

Name Dog 1 edited

Facts Dogs rely a lot on their sense of smell to find food, potential dangers, and friends, so needless to say they sniff a lot. Their noses are designed so smells can stay in their nose while air can move in and out of their lungs at the same time, which means they can breathe freely and still work out what that smell is!

Edit Pet

Delete Pet

Items

SELECT * FROM c Edit Filter

id	/_partitionKey
712207b2-a433-340...	

Load more


```
{
    "id": "712207b2-a433-340c-e023-db1ce609645b",
    "name": "Dog 1 edited",
    "facts": "Dogs rely a lot on their sense of smell to find food, potential dangers, and friends, so needless to say they sniff a lot. Their nos
    "category": "dog",
    "imagename": "dog1.jpg",
    "_rid": "ZMYkAJTIBIsLAAAAAAAAAA==",
    "_self": "dbs/ZMYkAA==/colls/ZMYkAJTIBIs=/docs/ZMYkAJTIBIsLAAAAAAAAAA==/",
    "_etag": "\"6300b796-0000-0c00-0000-6161cecb0000\"",
    "_attachments": "attachments/",
    "_ts": 1633799883
}
```


Hello!

This is a Cloud Development Homework 3 project Zoo-Shop web-site by Zhuman Rakhat

Code powered by Node JS, Bootstrap and cloud technologies Azure like Cosmos DB and Blob Storages

My github page

Dog 1 edited

Dogs rely a lot on their sense of smell to find food, potential dangers, and friends, so needless to say they sniff a lot. Their noses are designed so smells can stay in their nose while air can move in and out of their lungs at the same time, which means they can breathe freely and still work out what that smell is!

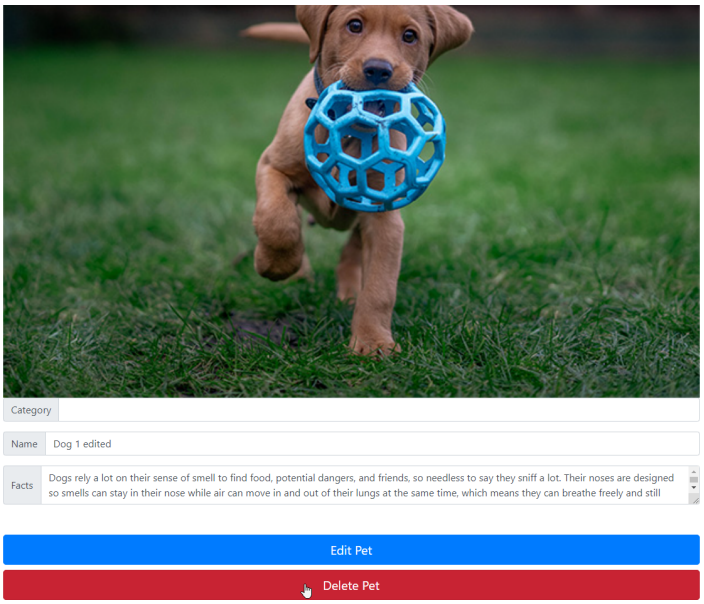
View details

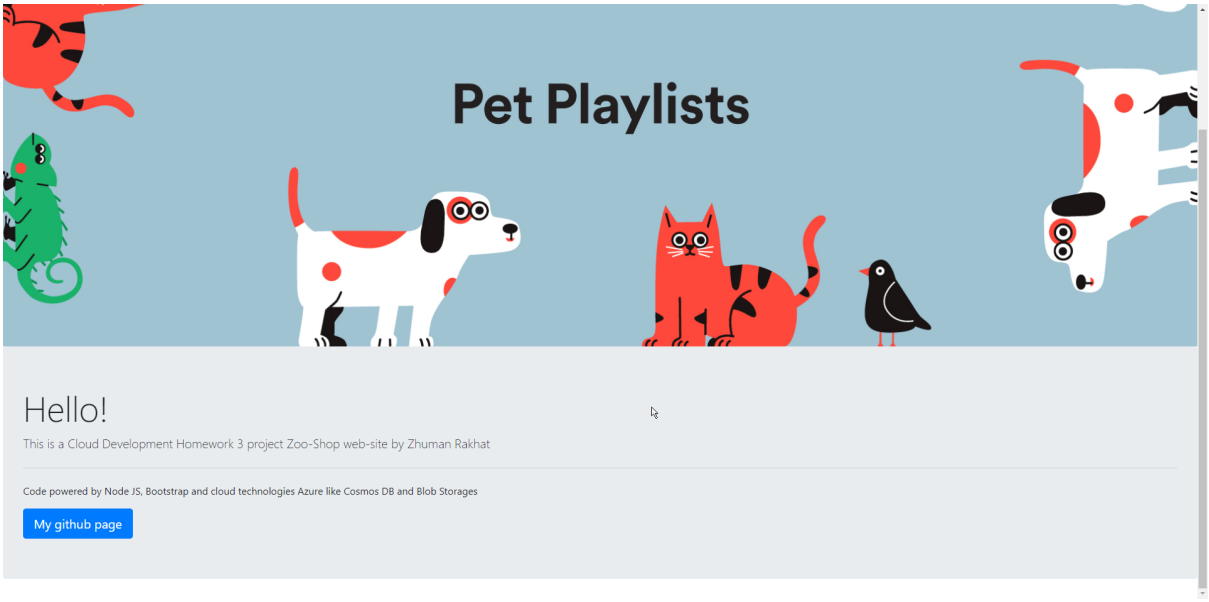
/Remove

```
async function deletePet(id){
    const { result: results } = await client.database(databaseId).container(containerId).item(id).delete();
    return results
}
```

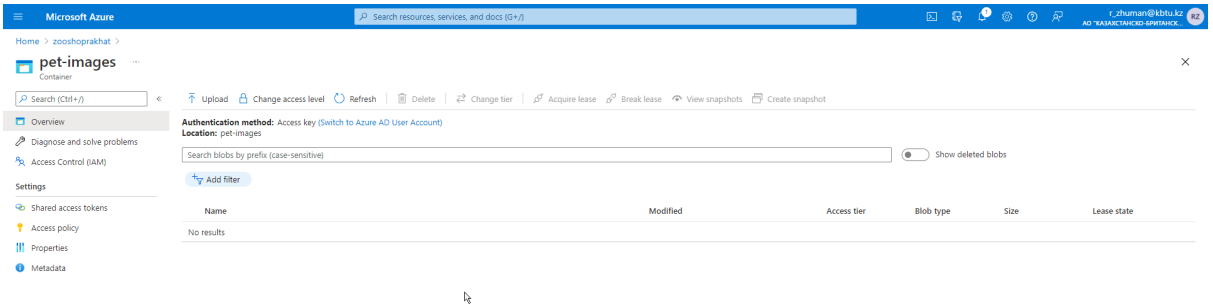
```
app.post('/edit', async function(req,res) {
    console.log('Edit ',req.body)
    const id = req.body.id
    const name = req.body.name
    const facts = req.body.facts
    const category = req.body.category
    const imagename = req.body.imagename
    const method = req.body.method

    if(method==='edit'){
        await updatePet(id, name, facts, category, imagename)
        res.render('edited')
    } else if (method === 'delete'){
        const deleted = await deletePet(id)
        await deleteImage(imagename)
        console.log(deleted)
        res.render('deleted')
    } else{
        res.status(400).render('index')
```

```
async function deleteImage(image){
    const containerClient = await getContainer()
    try{
        containerClient.deleteBlob(image)
    } catch {
        console.log("Can not find image by blob name ", image)
    }
}
```

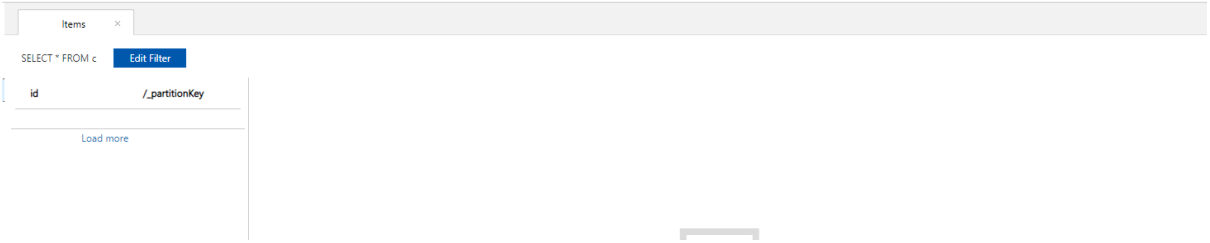




In blob storage deleted



Also in cosmos db



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

```
AZURE_STORAGE_CONNECTION_STRING=oHYs4OjNK1vu7CGn8VkDkqJCFD+U8xVzmR0j123xCAH4K5FlHJBQeDDqwgB36W6HBE92kbgoLaFzkpE/W6rrIQ==
COSMOS_ENDPOINT=https://zooshopdatastorage.documents.azure.com:443/
COSMOS_KEY=LmcUs0eULPDsf0NyyhhAqoLElVkcwqKL5iTSB5RhO2R1XjzCtcMUHd8lQn0QiHNlUnlzDOL6kTjb58uC083RqA==
COSMOS_DATABASE_ID=zooshopdb
COSMOS_CONTAINER_ID=zooshopcontainer
```

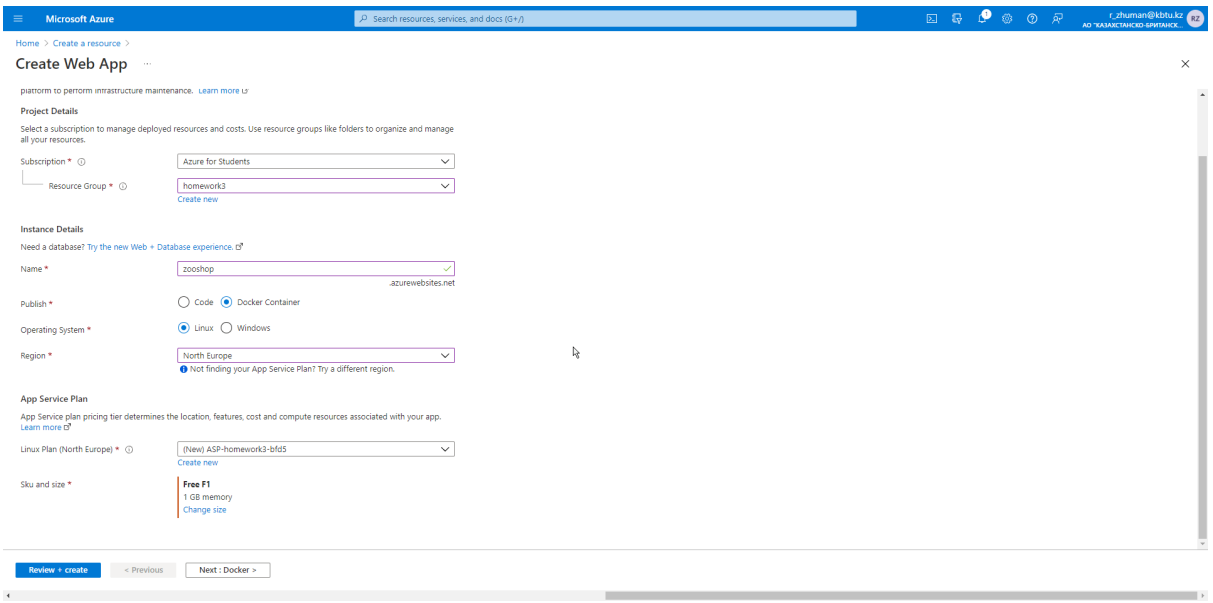
Build & push

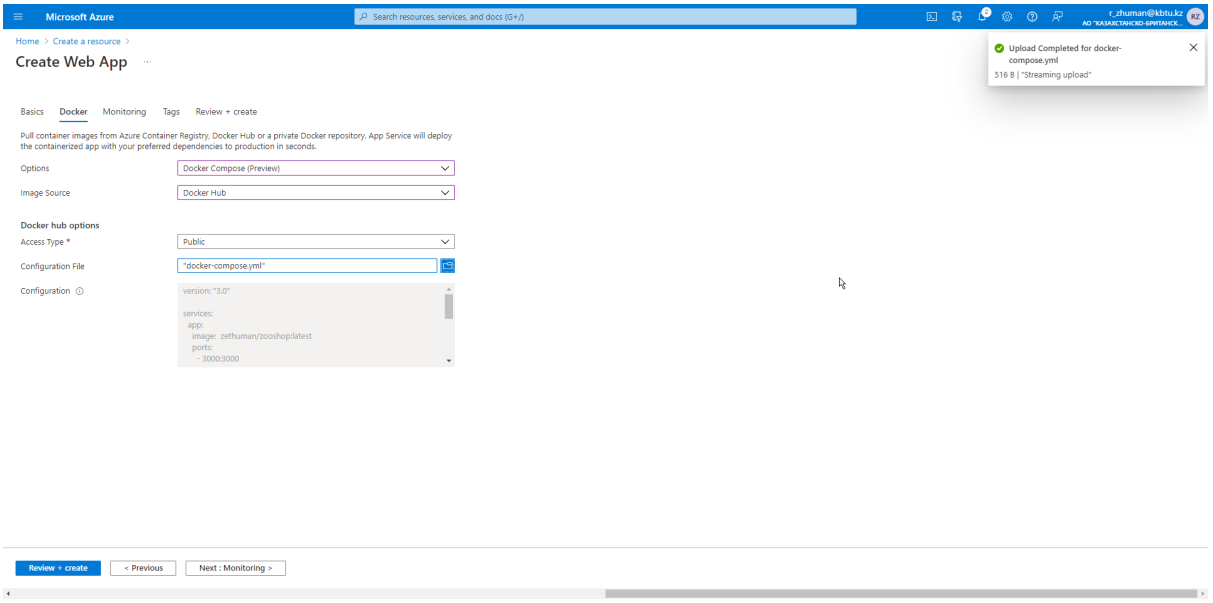
```
D:\Source\NodeJS\Azure\HW3>docker build -t zooshop:latest .
[+] Building 11.5s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                                                                0.0s
 => => transferring dockerfile: 31B                                                                                 0.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 34B                                                                                    0.0s
 => [internal] load metadata for docker.io/library/node:14-alpine                                                   2.8s
 => [auth] library/node:pull token for registry-1.docker.io                                                         0.0s
 => [1/4] FROM docker.io/library/node:14-alpine@sha256:6e52e0b3bedfb494496488514d18bee7fd503fd4e44289ea012ad02f8f41a312  0.0s
 => => resolve docker.io/library/node:14-alpine@sha256:6e52e0b3bedfb494496488514d18bee7fd503fd4e44289ea012ad02f8f41a312  0.0s
 => [internal] load build context                                                                                   3.6s
 => => transferring context: 10.05MB                                                                                3.6s
 => CACHED [2/4] WORKDIR /usr/src/app                                                                               0.0s
 => [3/4] COPY . /usr/src/app                                                                                       1.0s
 => [4/4] RUN npm install                                                                                           3.3s
 => exporting to image                                                                                              0.6s
 => => exporting layers                                                                                             0.6s
 => => writing image sha256:02ae308c74d1b93cae9100bcc954dc472bd5e89d56984755905ea44a9ec830b8                        0.0s
 => => naming to docker.io/library/zooshop: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\HW3>docker tag zooshop:latest zethuman/zooshop:latest
```

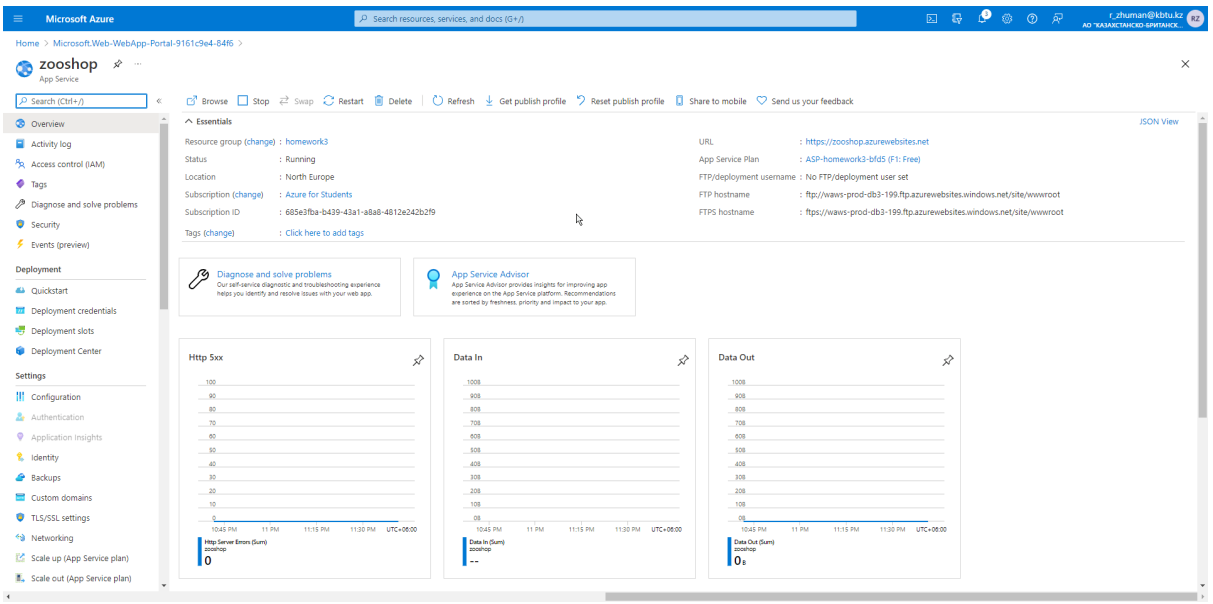
docker-compose

```
version: "3.0"

services:
  app:
    image:  zethuman/zooshop:latest
    ports:
      - 3000:3000
    environment:
      AZURE_STORAGE_CONNECTION_STRING: oHYs4OjNK1vu7CGn8VkDkqJCFD+U8xVzmR0j123xCAH4K5FlHJBQeDDqwgB36W6HBE92kbgoLaFzkpE/W6rrIQ==
      COSMOS_ENDPOINT: https://zooshopdatastorage.documents.azure.com:443/
      COSMOS_KEY: LmcUs0eULPDsf0NyyhhAqoLElVkcwqKL5iTSB5RhO2R1XjzCtcMUHd8lQn0QiHNlUnlzDOL6kTjb58uC083RqA==
      COSMOS_DATABASE_ID: zooshopdb
      COSMOS_CONTAINER_ID: zooshopcontainer
```

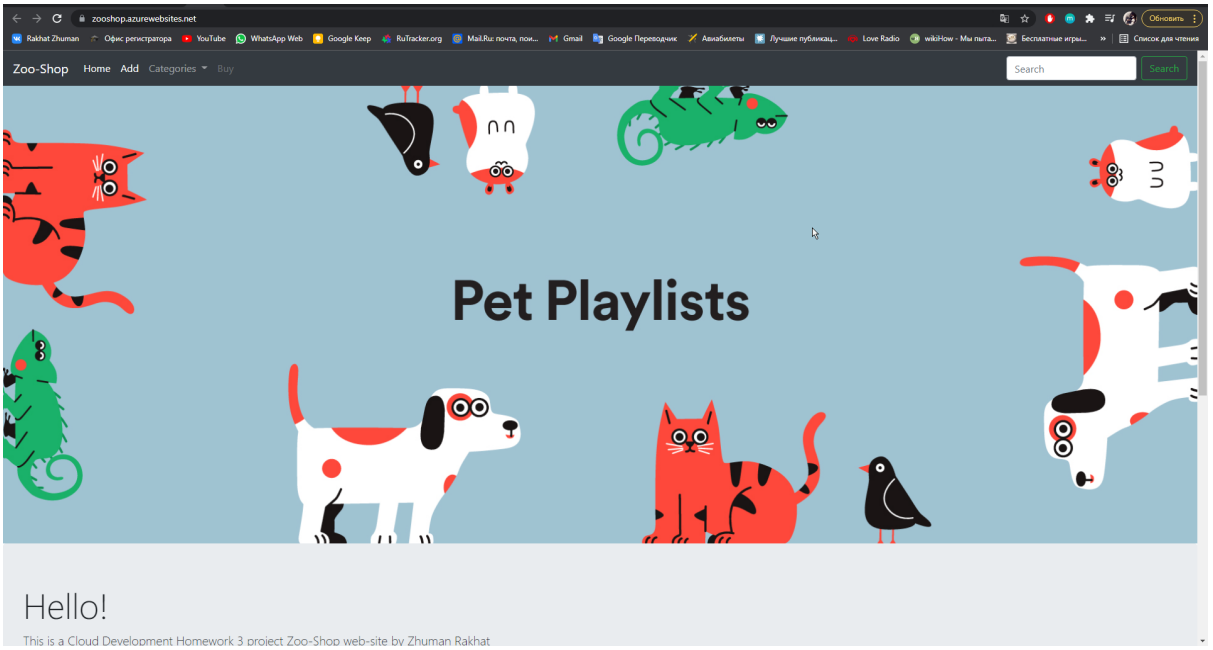
Create Web App


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Create a resource >
Create Web App
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 *
homework3
Create new
Instance Details
Need a database? Try the new Web + Database experience.
Name *
zooshop
.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-homework3-bfd5
Create new
Sku and size *
Free F1
1 GB memory
Change size
Review + create
< Previous
Next : Docker >

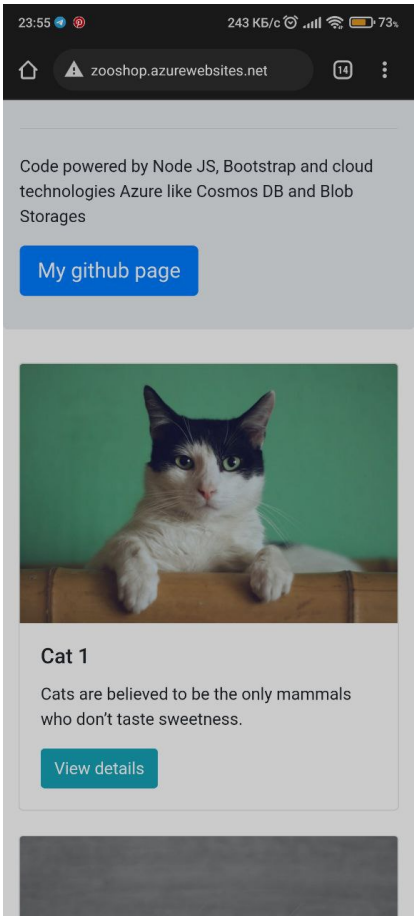

Microsoft Azure
Search resources, services, and docs (G+/)
Home > Create a resource >
Create Web App
Upload Completed for docker-compose.yml
516 B | "Streaming upload"
Basics
Docker
Monitoring
Tags
Review + create
Pull container images from Azure Container Registry, Docker Hub or a private Docker repository. 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/zooshop:latest
ports:
- 3000:3000
Review + create
< Previous
Next : Monitoring >

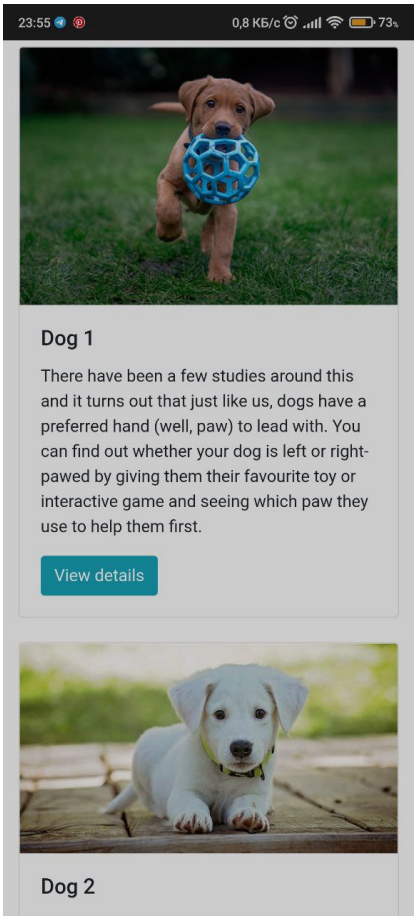

Microsoft Azure
Search resources, services, and docs (G+/)
Home > Microsoft.Web-WebApp-Portal-9161c9e4-84f6 >
zooshop
App Service
Search (Ctrl+/)
Browse
Stop
Swap
Restart
Delete
Refresh
Get publish profile
Reset publish profile
Share to mobile
Send us your feedback
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Security
Events (preview)
Deployment
Quickstart
Deployment credentials
Deployment slots
Deployment Center
Settings
Configuration
Authentication
Application Insights
Identity
Backups
Custom domains
TLS/SSL settings
Networking
Scale up (App Service plan)
Scale out (App Service plan)
Essentials
JSON View
Resource group (change) : homework3
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://zooshop.azurewebsites.net
App Service Plan : ASP-homework3-bfd5 (F1: Free)
FTP/deployment username : No FTP/deployment user set
FTP hostname : ftp://waws-prod-db3-199.ftp.azurewebsites.windows.net/site/wwwroot
FTPS hostname : ftps://waws-prod-db3-199.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.
Http 5xx
Data In
Data Out
0
--
0 B

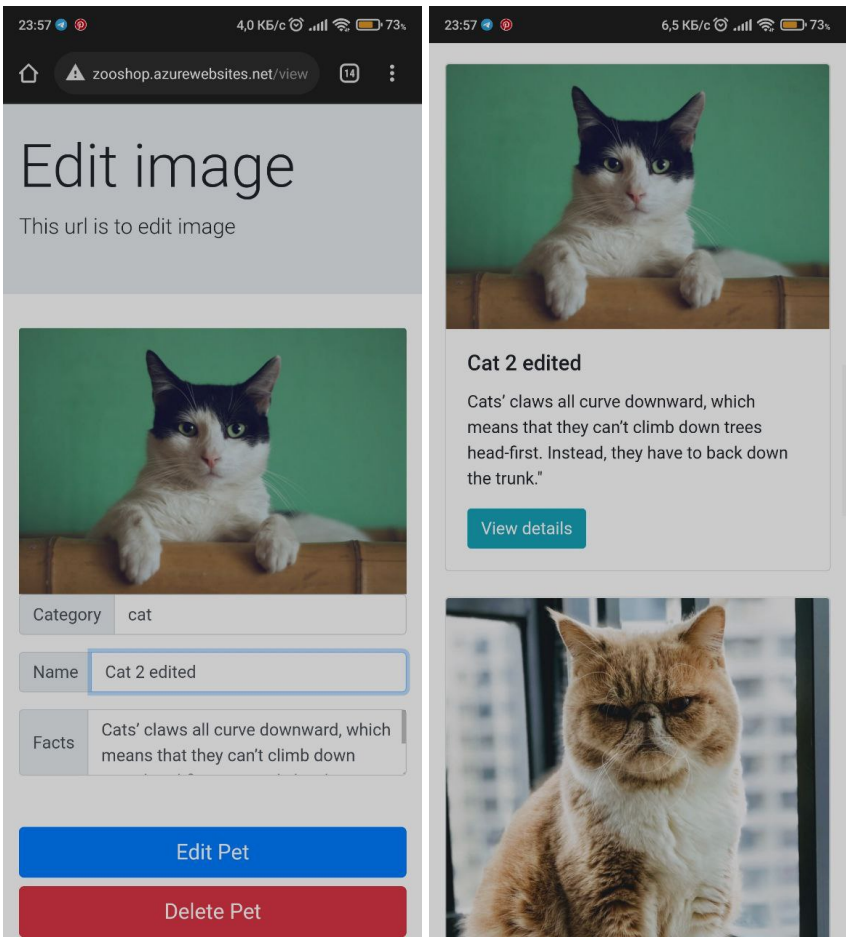
Result



Check from phone







Cleaning up workspace