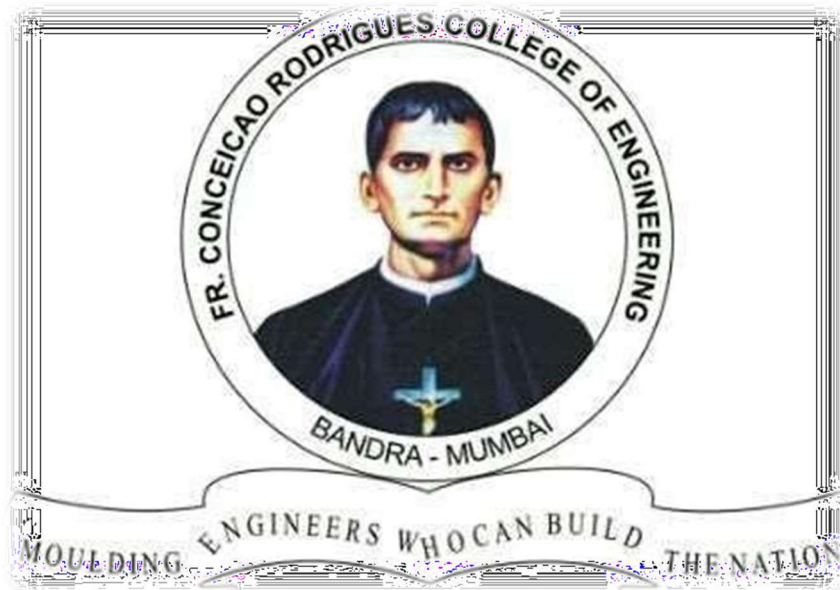


Fr. Conceicao Rodrigues College of Engineering
Bandra (West), Mumbai 400050



Cloud Computing

Department of Computer Engineering

A project report on

FILE MANAGEMENT SYSTEM

Prepared by:

8944 Abhi Gupta

8940 Warren Fernandes

8949 Vinyas Kulal

8950 Liny Mathew

Supervised by:

Prof. Ankita Amburle

Department of Computer Eng.

April 2022

Department Of Computer Engineering

Academic Term Jan-Apr 2022

Class : TE Computer – B (Sem VI)

Subject : Cloud Computing

Title of the Project	FILE MANAGEMENT SYSTEM
Name	8944 Abhi Gupta 8940 Warren Fernandes 8949 Vinyas Kulal 8950 Liny Mathew
Date Of Performance	22/04/2022
Date Of Submission	22/04/2022
Batch	B

Evaluation:

Sr. No	Rubric	Grade
1	Timeline(2)	
2	Completeness(5)	
3	Project specific Features (8)	
4	Total (15)	

Signature of Teacher:

FILE MANAGEMENT SYSTEM

Title: To build a File Management System using Cloud Computing.

Abstract:

The File Management System is used to upload, maintain download files in order to minimize the number of hard copies on hand. Users will be able to manage their documents online as part of this initiative. The features of the project are possibly inspired by Google Drive File Management, with the exception that this framework would only concentrate on a single entity or institution.

A file management system helps you to store, exchange, and maintain all of your company's/organization's file records. This project's functionality can be applied to colleges, offices, or any other organization that wants to handle and store their file records using cloud service

Introduction:

Using File Management system users can upload and download files, documents, images, videos through online in a secured manner. In this user can upload/download documents and files in any format. It's an online web application where all the file details store in Azure Storage, Azure Container Registry to store the docker image, Azure App Service for PaaS and Azure App Service Plan for IaaS which creates an EC2 instance on computer.

Objective: Designing a file management system hosted on public cloud platform (Azure) that covers the concept of IaaS, Pass and SaaS services including Docker.

Video Link: View Demonstration -

https://drive.google.com/drive/folders/1nSRPrEYmQIGk4UBA4q_kjFffLEvh7jPX?usp=sharing

Operations:

- Azure Storage Blob Library
- Azure Storage - SaaS
- Azure Container Registry to store the docker image
- Azure App Service for PaaS
- Azure App Service Plan for IaaS which creates an EC2 instance on computer.

Implementation:

Step 1: Azure Services for Storage –

The Azure Storage platform is Microsoft's cloud storage solution for modern data storage scenarios. Azure Storage offers highly available, massively scalable, durable, and secure storage for a variety of data objects in the cloud. Azure Storage data objects are accessible from anywhere in the world over HTTP or HTTPS via a REST API.

As in our application, files have been stored, retrieved and downloaded, we use Azure Blob storage library.

Azure Blob Storage is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured data, such as text or binary data. Here, we create a storage account, input subscription, its resource group, storage name – ***miniprojectccdb***, its region performance, redundancy. Once reviewed, a storage account will be created.

To store files, we create blob containers.

NAME	ACCESS TIER	ACCESS TIER LAST MODIFIED	LAST MODIFIED	BLOB TYPE	CONTENT TYPE	SIZE	STATUS	REMAINING
2112.01237.pdf	Hot (inferred)		4/17/2022, 8:21:12 PM	Block Blob	application/octet-stream	1.7 MB	Active	
8940_MC_Exp19.pdf	Hot (inferred)		4/17/2022, 11:36:07 PM	Block Blob	application/octet-stream	727.8 KB	Active	
8940_Warren_IAM_CC.pdf	Hot (inferred)		4/18/2022, 1:13:10 AM	Block Blob	application/octet-stream	688.1 KB	Active	
8950_Liny_Docker_CC.pdf	Hot (inferred)		4/17/2022, 8:47:55 PM	Block Blob	application/octet-stream	10.1 MB	Active	
8950_Liny_Exp12_CC.pdf	Hot (inferred)		4/17/2022, 9:06:35 PM	Block Blob	application/octet-stream	1.2 MB	Active	
Clarification Regarding Term Calendar.pdf	Hot (inferred)		4/18/2022, 7:41:41 AM	Block Blob	application/octet-stream	356.3 KB	Active	

Showing 1 to 6 of 6 cached items

Microsoft Azure Search resources, services, and docs (G+/)

Home > storage-res >

miniprojdb

Storage account

Search (Ctrl+/)

Open in Explorer Delete Move Refresh Mobile Feedback

Overview

- Activity log
- Tags
- Diagnose and solve problems
- Access Control (IAM)
- Data migration
- Events
- Storage browser (preview)

Data storage

- Containers
- File shares
- Queues
- Tables

Essentials

Resource group (move) : storage-res

Location : Southeast Asia

Primary/Secondary Location : Primary: Southeast Asia, Secondary: East Asia...

Subscription (move) : Azure for Students

Subscription ID :

Disk state : Primary: Available, Secondary: Available

Tags (edit) : Click here to add tags

Performance : Standard

Replication : Read-access geo-redundant storage (RA-GRS)

Account kind : StorageV2 (general purpose v2)

Provisioning state : Succeeded

Created : 4/15/2022, 11:51:10 PM

JSON View

Properties Monitoring Capabilities (7) Recommendations Tutorials Developer Tools

Blob service

Hierarchical namespace Disabled

Default access tier Hot

Blob public access Enabled

Security

Require secure transfer for REST API operations Enabled

Storage account key access Enabled

Minimum TLS version Version 1.2

Microsoft Azure Search resources, services, and docs (G+/)

Home >

storage-res

Resource group

Search (Ctrl+/)

Create Manage view Delete resource group Refresh Export to CSV Open query Assign tags Move

Overview

- Activity log
- Access control (IAM)
- Tags
- Resource visualizer
- Events

Settings

- Deployments
- Security
- Policies
- Properties
- Locks

Cost Management

Essentials

Subscription (move) Azure for Students

Subscription ID :

Location Southeast Asia

Deployments 1 Succeeded

Tags (edit) Click here to add tags

Resources Recommendations

Filter for any field... Type == all Location == all Add filter

Showing 1 to 1 of 1 records. Show hidden types No grouping List view

Name	Type	Location
miniprojdb	Storage account	Southeast Asia

Step 2: Streamlit application –

Streamlit is an open-source app framework for Machine Learning and Data Science teams. Create beautiful web apps in minutes. We use python scripts to build this application.

The main functions are –

- Uploading a file using Browse button
- Downloading a file using Download button
- Displaying all the list of files uploaded in the download section.

```
main.py > ...
1  import streamlit as st
2  from azure.storage.blob import BlobClient, ContainerClient
3
4  def upload():
5      st.title("Cloud Computing Lab - Mini Project")
6      connection_string = "DefaultEndpointsProtocol=https;AccountName=cloudcomputinglab;AccountKey=...;BlobEndpoint=https://cloudcomputinglab.blob.core.windows.net/;QueueEndpoint=https://cloudcomputinglab.queue.core.windows.net/;TableEndpoint=https://cloudcomputinglab.table.core.windows.net/"
7      uploaded_file = st.file_uploader('', type= '.pdf', accept_multiple_files=False)
8      if st.button("Upload"):
9          blob_client = BlobClient.from_connection_string(connection_string, container_name="photos", blob_name=uploaded_file.name)
10         blob_client.upload_blob(uploaded_file.read())
11         st.sidebar.info("Successfully Uploaded")
12
13  def download():
14      st.title("Cloud Computing Lab - Mini Project")
15      connection_string = "DefaultEndpointsProtocol=https;AccountName=cloudcomputinglab;AccountKey=...;BlobEndpoint=https://cloudcomputinglab.blob.core.windows.net/;QueueEndpoint=https://cloudcomputinglab.queue.core.windows.net/;TableEndpoint=https://cloudcomputinglab.table.core.windows.net/"
16      container_client = ContainerClient.from_connection_string(conn_str=connection_string, container_name="photos")
17      blobs = container_client.list_blobs()
18      for blob in blobs:
19          st.markdown(blob.name)
20          if st.button("Download", key=blob.name):
21              blob_client = BlobClient.from_connection_string(connection_string, container_name="photos", blob_name=blob.name)
22              bytes = blob_client.download_blob().readall()
23              st.download_button('Download file', bytes)
24
25  if __name__ == '__main__':
26      nav = st.sidebar.radio("", ["Upload", "Download"])
27      if nav == "Upload":
28          upload()
29      if nav == "Download":
30          download()
```

We get a connection string using Azure Blob. Using the string we upload a file of .pdf type. With the help of Blob Client, the files have been uploaded/retrieved into/from the azure storage

The downloaded function consists of downloading a file as well as displaying all the files name stored using blob client.

Step 3: Deploy the application using Paas.

For deploying the application, we need to containerize the app using docker.

Dockerfile is a text file which contains all the steps to be performed to create our application as a image.

```
Dockerfile > ...
1  FROM python:3.8-slim-buster
2  COPY . /app
3  WORKDIR /app
4  RUN apt-get update && \
5      apt-get -y install sudo
6  RUN pip install -r requirements.txt
7  EXPOSE 8501
8  ENTRYPOINT ["streamlit","run"]
9  CMD ["main.py"]
```

```
requirements.txt
1  click==8
2  azure-storage-blob==12.11.0
3  streamlit==1.8.1
```

The base image i.e. python 3.8, all the related files to be copied, the sudo command to install streamlit, a requirement.txt file which contains all the details that our application need to be executed, a port 8501 and the command to run streamlit main.py has been included into the docker file

The command *docker build -t name* will create an image.

Now to push this into container registry of azure, we need to use container registry service of azure

Azure Container Registry

Azure Container Registry is a **private registry service for building, storing, and managing container images and related artifacts**

For creating a container registry, the following needs to be inputted,

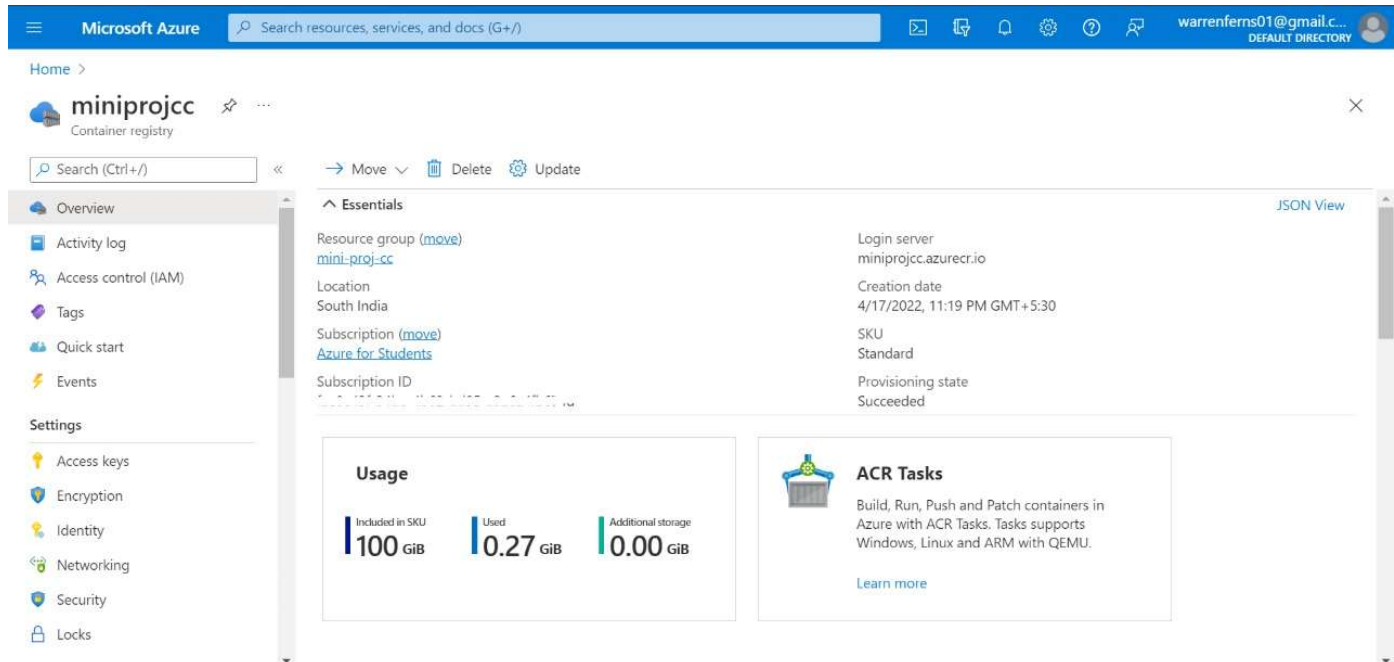
Subscription: Select your valid Azure subscription where you want to create container registry.

Resource group: Select the resource group which you created in previous step.

Registry name: Name of your Azure Container registry to be created.

Location: The location where you want to create ACR (Keep same as the resource group Region).

SKU: Select the registry tier Basic. (Available are Basic, Standard, Premium)



The screenshot displays the Azure Portal interface for a Container Registry named 'miniprojcc'. The left sidebar shows the navigation menu with options like Overview, Activity log, Access control (IAM), Tags, Quick start, Events, and Settings. The main content area is divided into two sections: 'Essentials' and 'Usage'. The 'Essentials' section lists the resource group (mini-proj-cc), location (South India), subscription (Azure for Students), and subscription ID. The 'Usage' section shows a bar chart with three bars: 'Included in SKU' at 100 GiB, 'Used' at 0.27 GiB, and 'Additional storage' at 0.00 GiB. The 'ACR Tasks' section provides information about building, running, pushing, and patching containers in Azure with ACR Tasks, supporting Windows, Linux, and ARM with QEMU. The top of the page shows the Microsoft Azure logo, a search bar, and the user's profile (warrenferns01@gmail.com).

Property	Value
Resource group	mini-proj-cc
Location	South India
Subscription	Azure for Students
Subscription ID	6a10f321-408d-4b31-b1e1-100000000000
Login server	miniprojcc.azurecr.io
Creation date	4/17/2022, 11:19 PM GMT+5:30
SKU	Standard
Provisioning state	Succeeded

Usage	Value
Included in SKU	100 GiB
Used	0.27 GiB
Additional storage	0.00 GiB

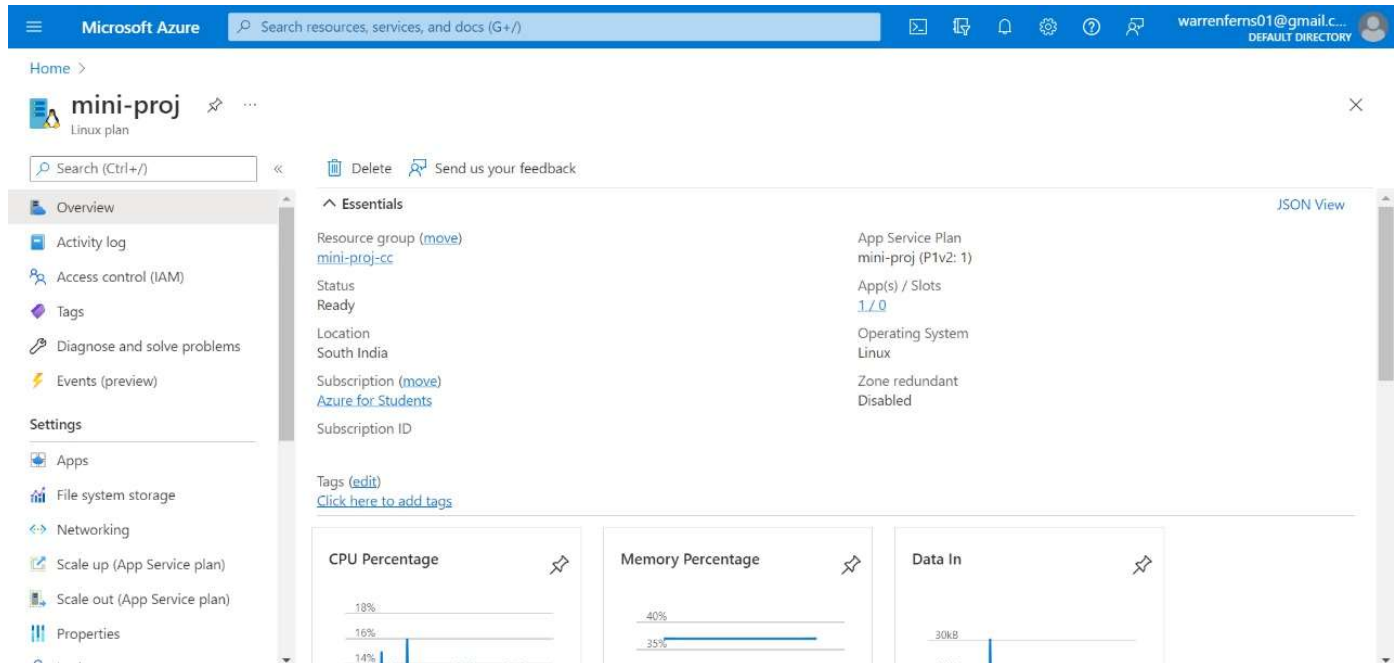
ACR Tasks
Build, Run, Push and Patch containers in Azure with ACR Tasks. Tasks supports Windows, Linux and ARM with QEMU.
[Learn more](#)

Here we push the docker file into the particular ACR.

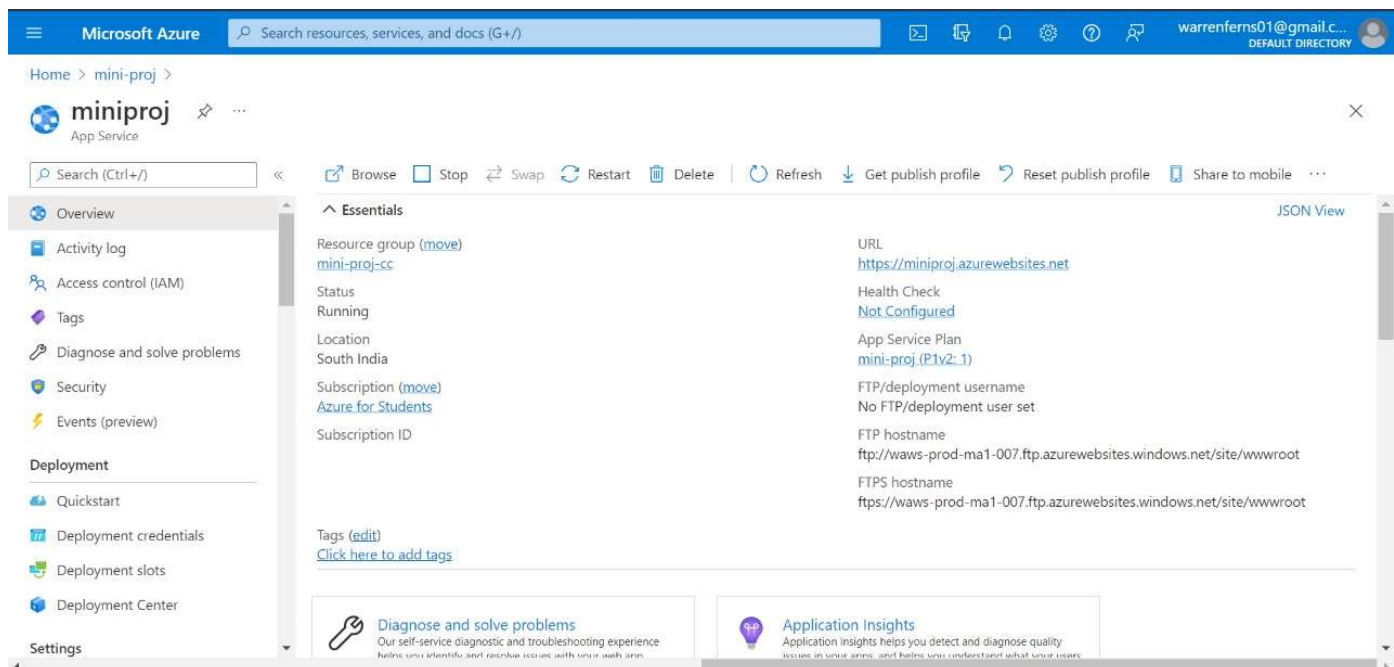
For deploying, Azure service plan is implemented.

An App Service plan defines a set of compute resources for a web app to run. The app service plan defines what specification of hardware your app runs on, and how many servers you have.

Once we create the app service, deploy the image to Azure service.
Here Iaas is being implement as a virtual machine has been created.



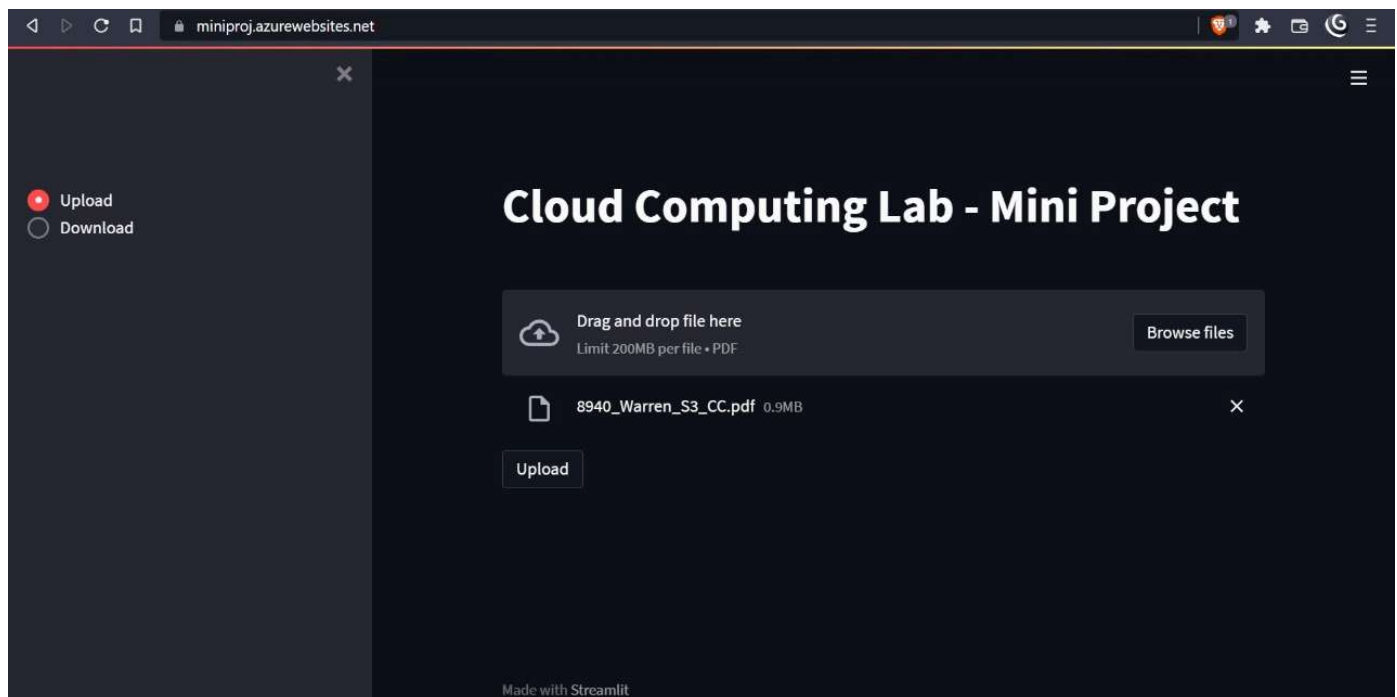
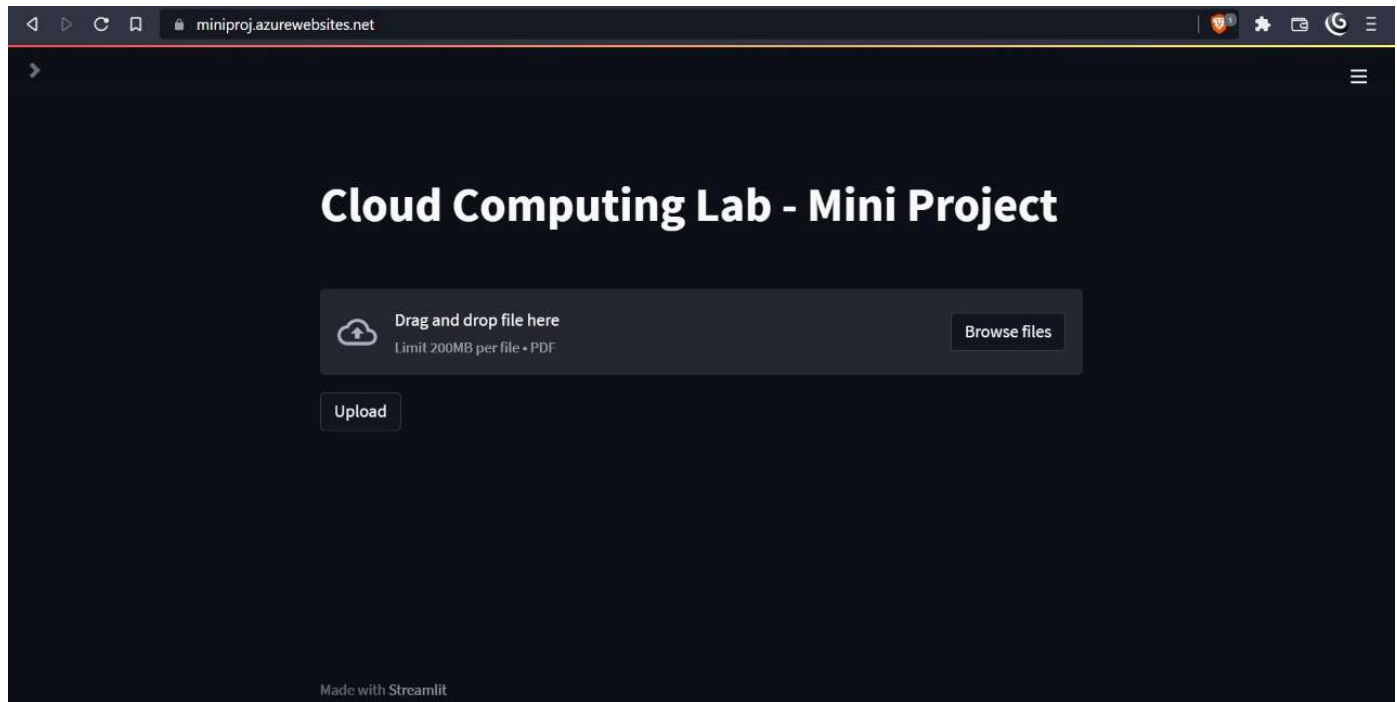
The screenshot shows the Microsoft Azure portal interface. At the top, there's a search bar and navigation icons. The main header indicates the user is logged in as 'warrenferns01@gmail.c...'. Below the header, the breadcrumb trail shows 'Home > mini-proj'. The resource 'mini-proj' is identified as a 'Linux plan'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events (preview), Settings, Apps, File system storage, Networking, Scale up (App Service plan), Scale out (App Service plan), and Properties. The main content area is titled 'Essentials' and lists key details: Resource group (mini-proj-cc), Status (Ready), Location (South India), Subscription (Azure for Students), and Subscription ID. To the right, a 'JSON View' tab is active, showing details like App Service Plan (mini-proj (P1v2: 1)), App(s) / Slots (1 / 0), Operating System (Linux), and Zone redundant (Disabled). Below these details are three charts: CPU Percentage (18%), Memory Percentage (40%), and Data In (30MB).

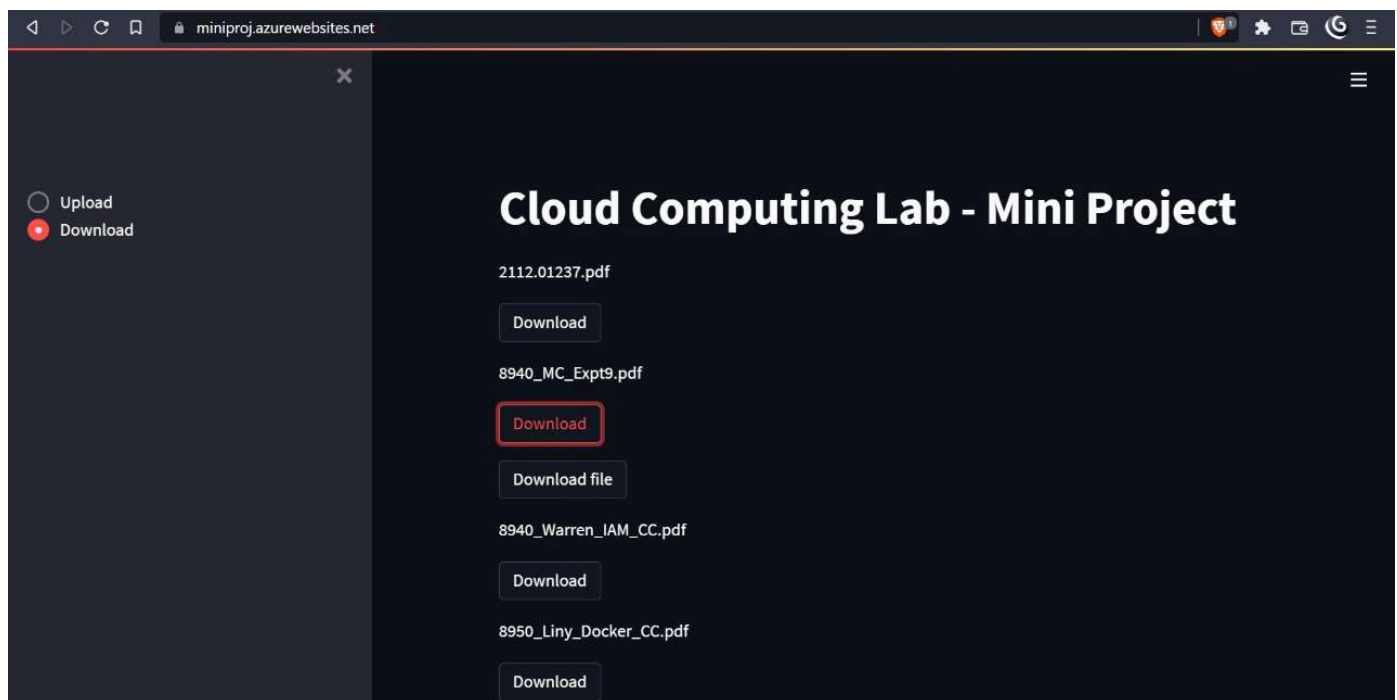


The screenshot shows the Microsoft Azure portal interface for an App Service resource named 'miniproj'. The breadcrumb trail is 'Home > mini-proj > miniproj'. The resource is identified as an 'App Service'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Security, Events (preview), Deployment, Quickstart, Deployment credentials, Deployment slots, and Deployment Center. The main content area is titled 'Essentials' and lists key details: Resource group (mini-proj-cc), Status (Running), Location (South India), Subscription (Azure for Students), and Subscription ID. To the right, a 'JSON View' tab is active, showing details like URL (https://miniproj.azurewebsites.net), Health Check (Not Configured), App Service Plan (mini-proj (P1v2: 1)), FTP/deployment username (No FTP/deployment user set), FTP hostname (ftp://waws-prod-ma1-007.ftp.azurewebsites.windows.net/site/wwwroot), and FTPS hostname (ftps://waws-prod-ma1-007.ftp.azurewebsites.windows.net/site/wwwroot). Below these details are two cards: 'Diagnose and solve problems' and 'Application Insights'.

The url mentioned specifies that our application has been hosted.

The application:





Conclusion:

A file management system used to store, upload and download a file has been implemented using Cloud computing that covers the concept of PaaS, IaaS and SaaS with the help of Docker.

References:

<https://docs.microsoft.com/en-us/azure/container-registry/container-registry-get-started-portal>

<https://docs.microsoft.com/en-us/azure/app-service/overview-hosting-plans>

<https://techcommunity.microsoft.com/t5/iis-support-blog/upload-and-download-files-from-azure-storage/ba-p/287834>

<https://azurelessons.com/upload-and-download-file-in-azure-blob-storage/>