# CLOUD APPLICATIONS DEVELOPMENT

## TOPIC: IMAGE RECOGNITION WITH IBM CLOUD VIRTUAL RECOGNITION

### PHASE 4

TEAM LEADER

S.RITHIK

TEAM MEMBERS

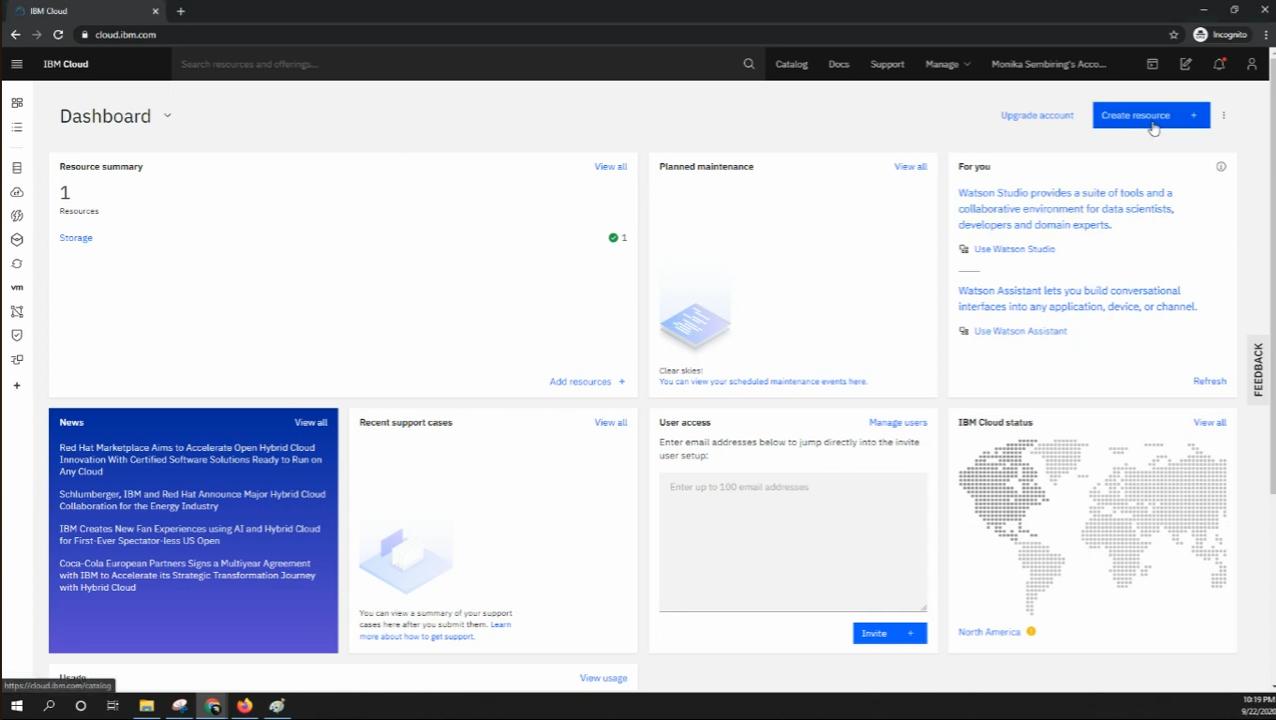
C.SUYAMBURAJ

G.KARTHICKRAJA

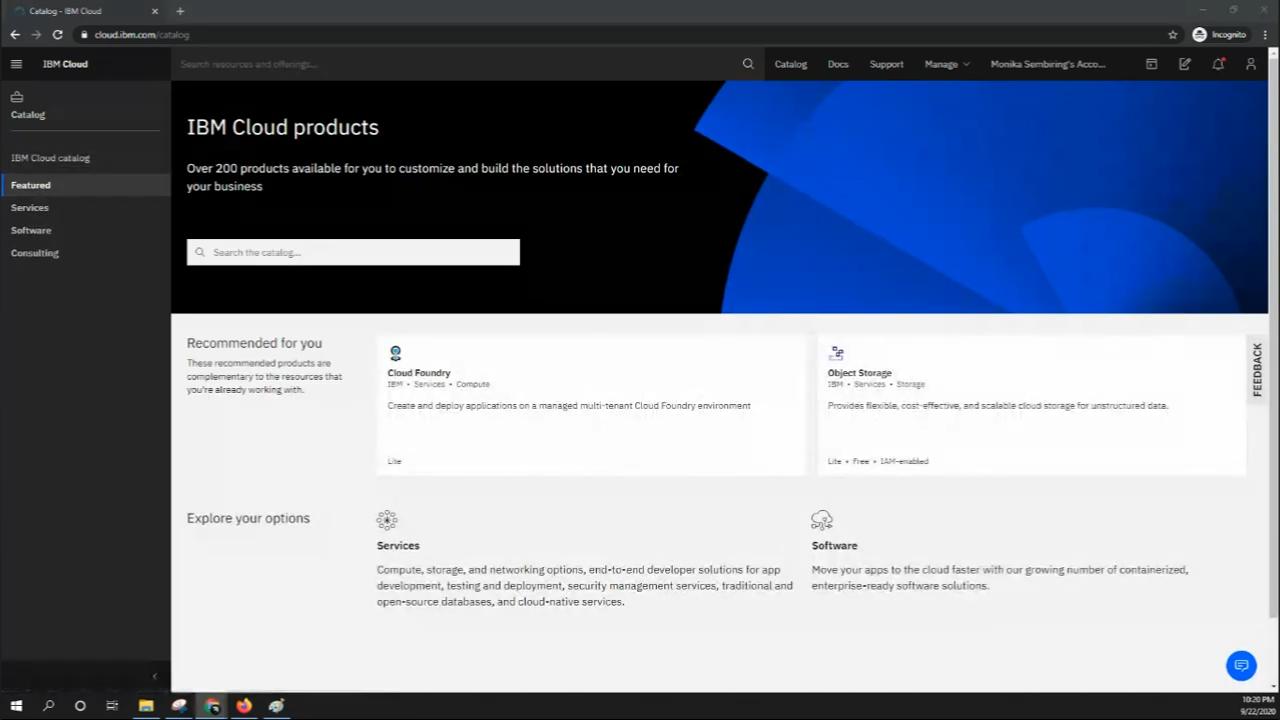
L.MOTHIR

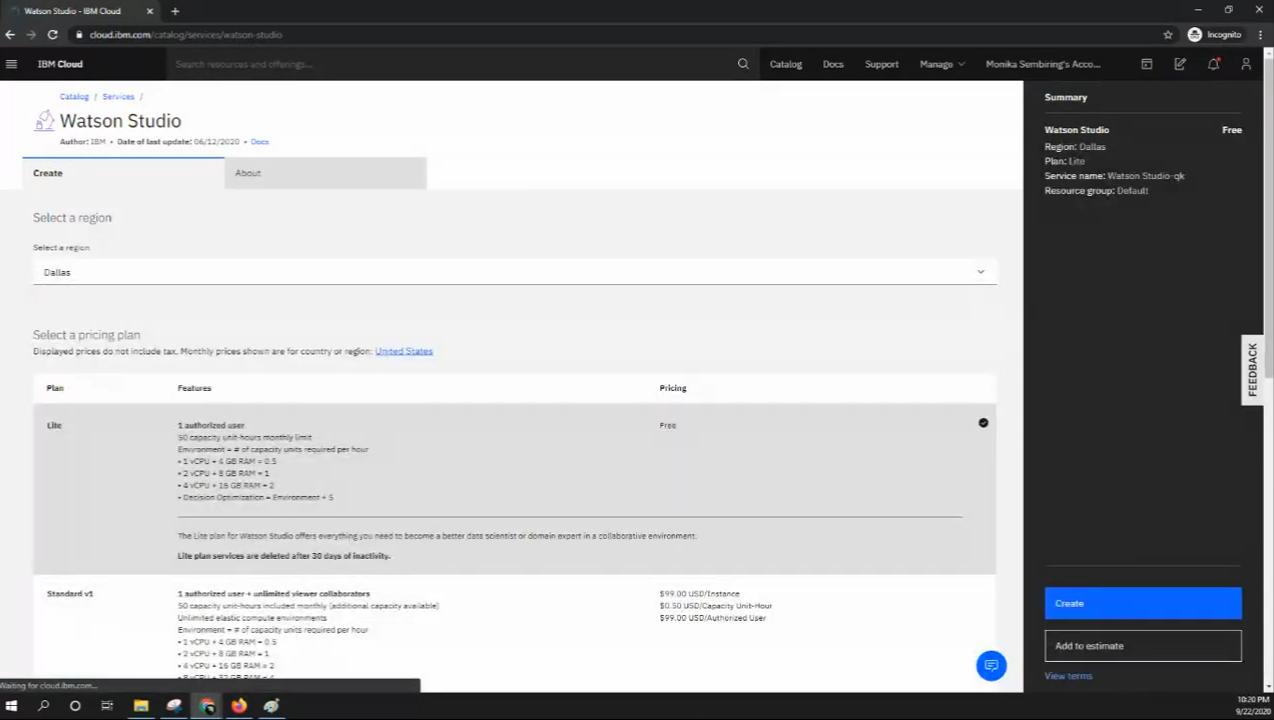
N.MADHANKUMAR

STEP 1 : OPEN IBM CLOUD DASHBOARD and Click CREATE RESOURCES

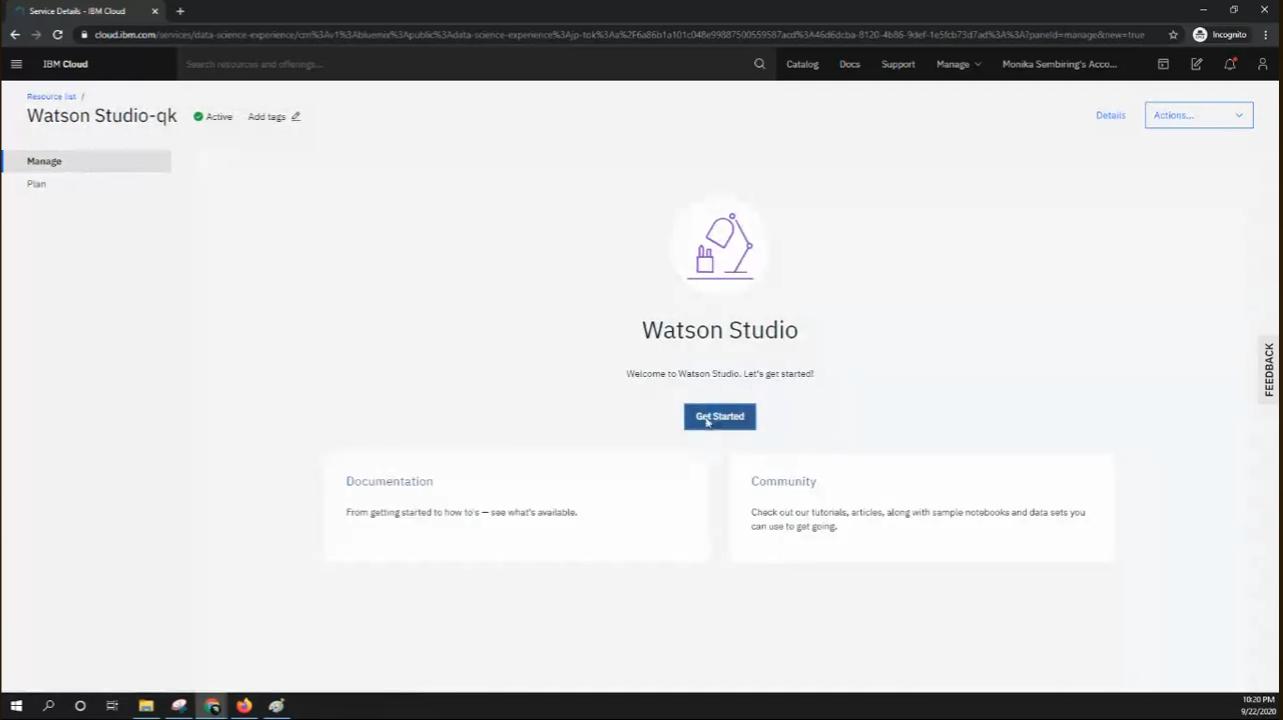


STEP 2 : After click CREATE RESOURCES It OPEN IBM CLOUD PRODUCTS

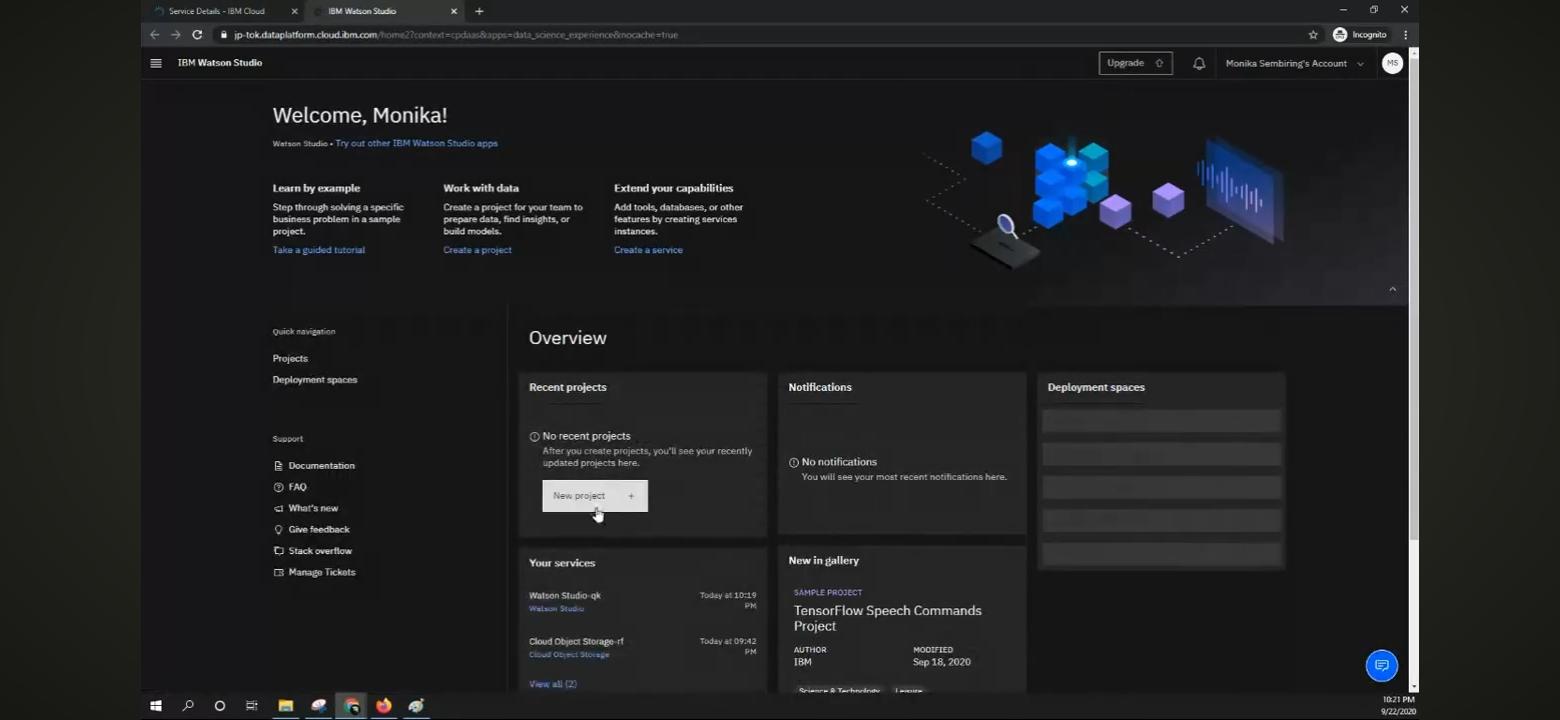


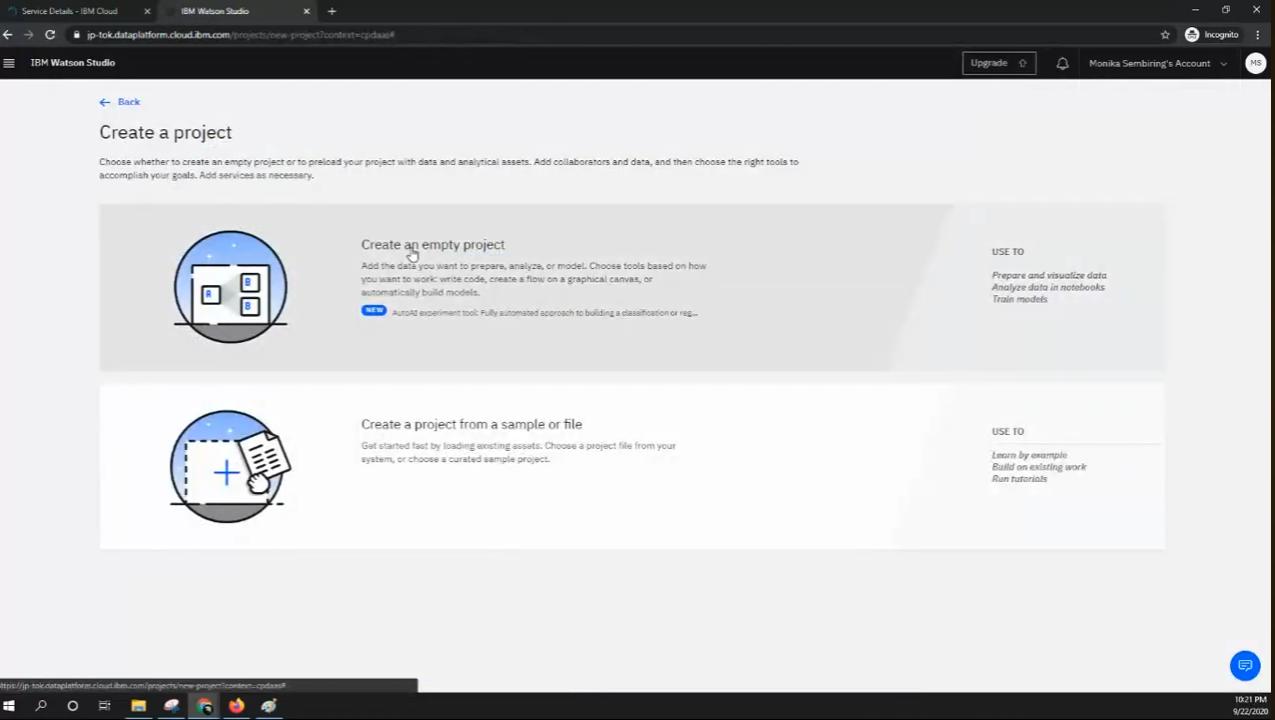
STEP 3 : IN WATSON STUDIO set REGION and Click CREATE

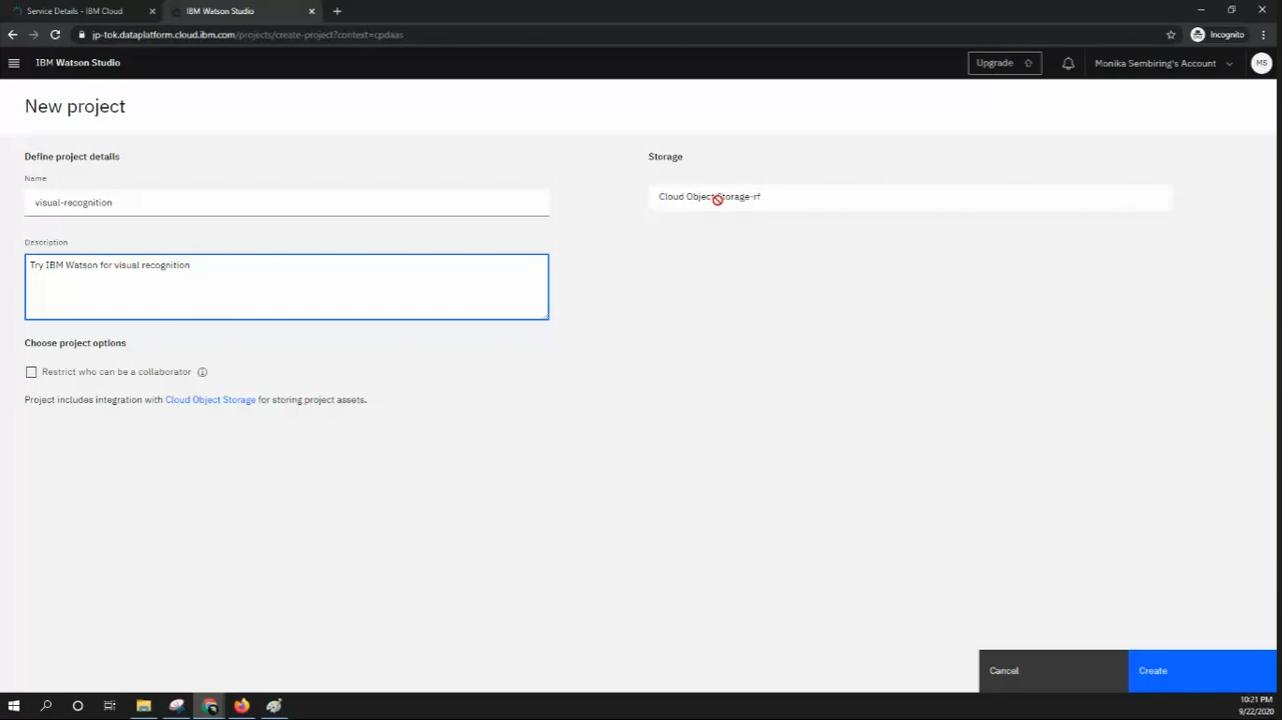
STEP 4 : THEN GET STARTED the Watson Studio

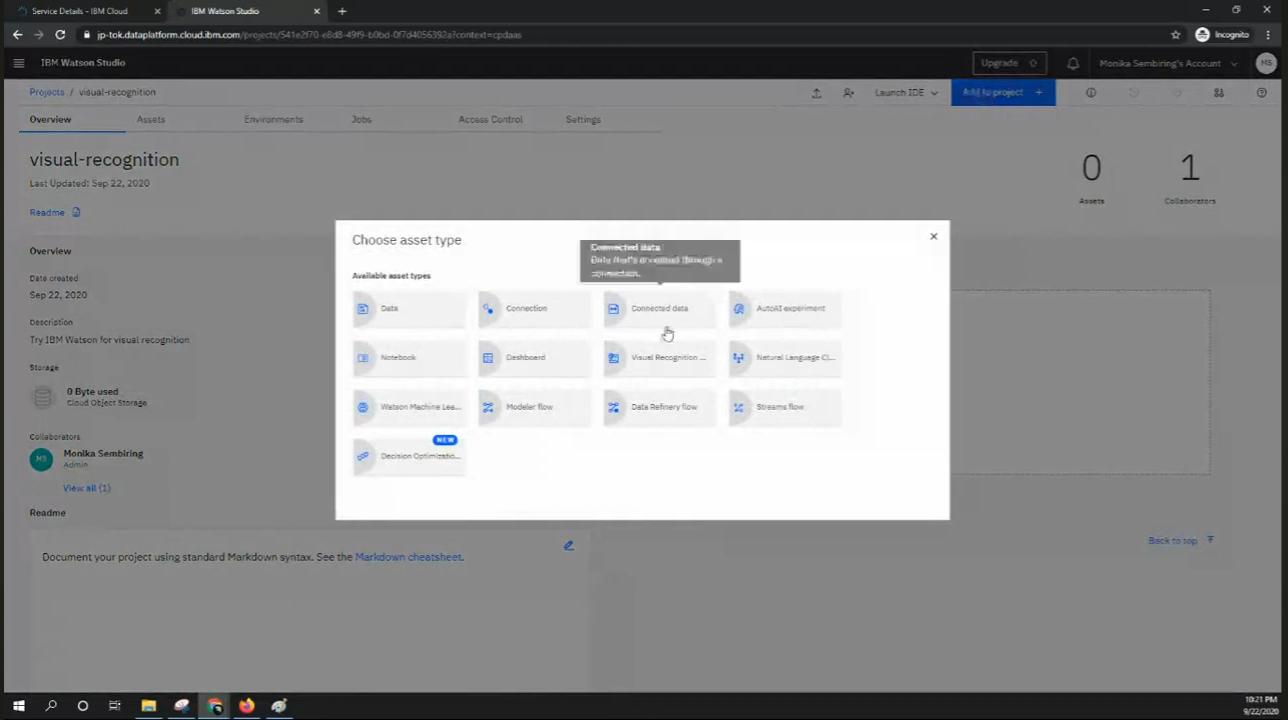


STEP 5 : CLICK NEW PROJECT

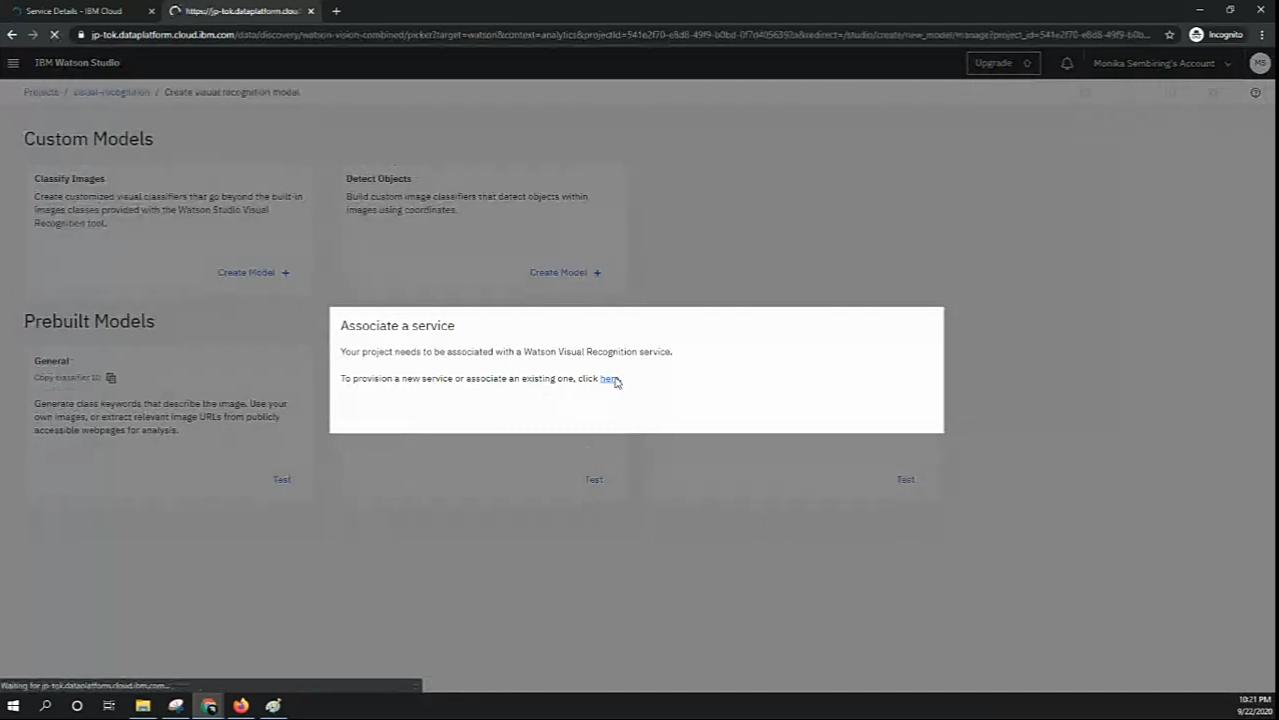
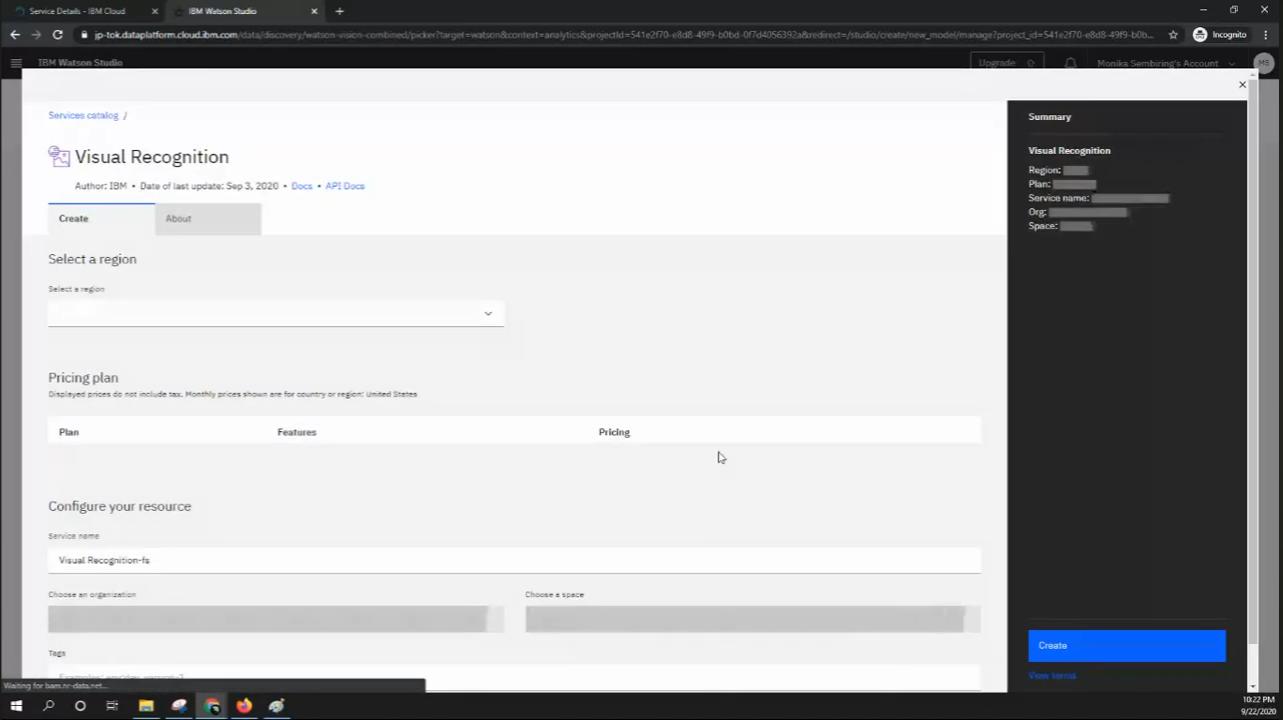


STEP 6 : After clicking NEW PROJECT and Click CREATE AN EMPTY PROJECT

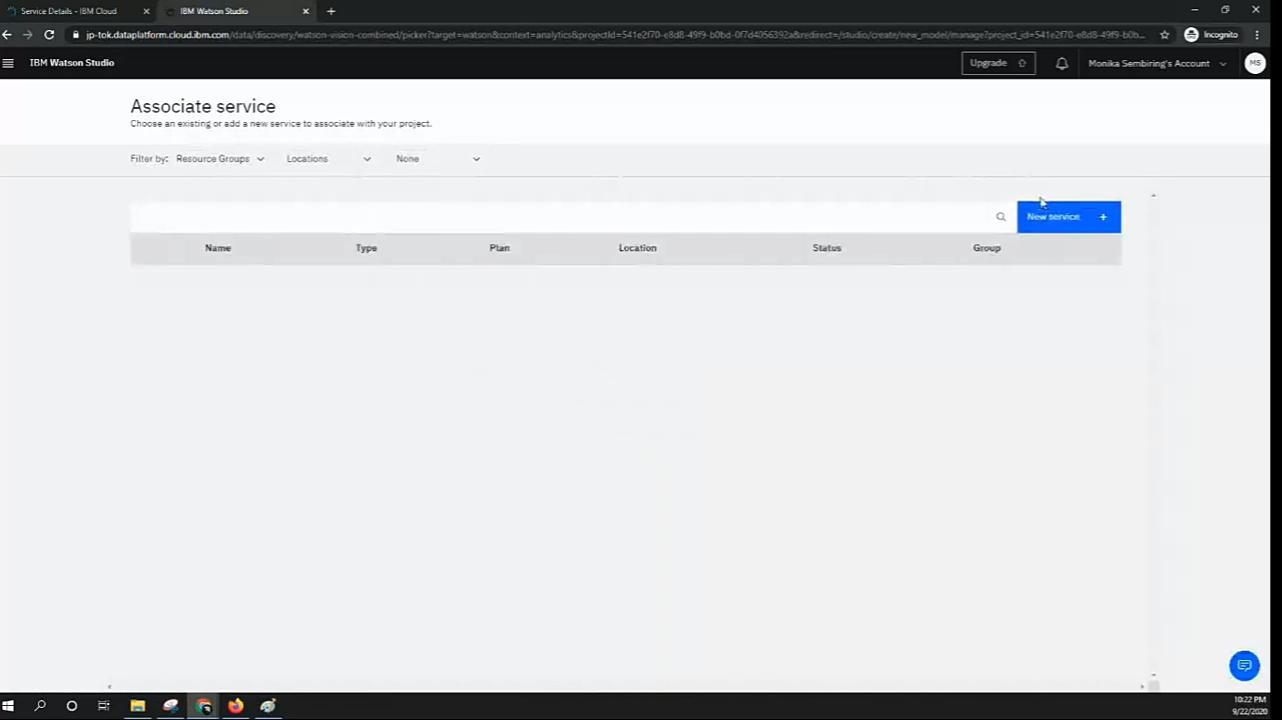


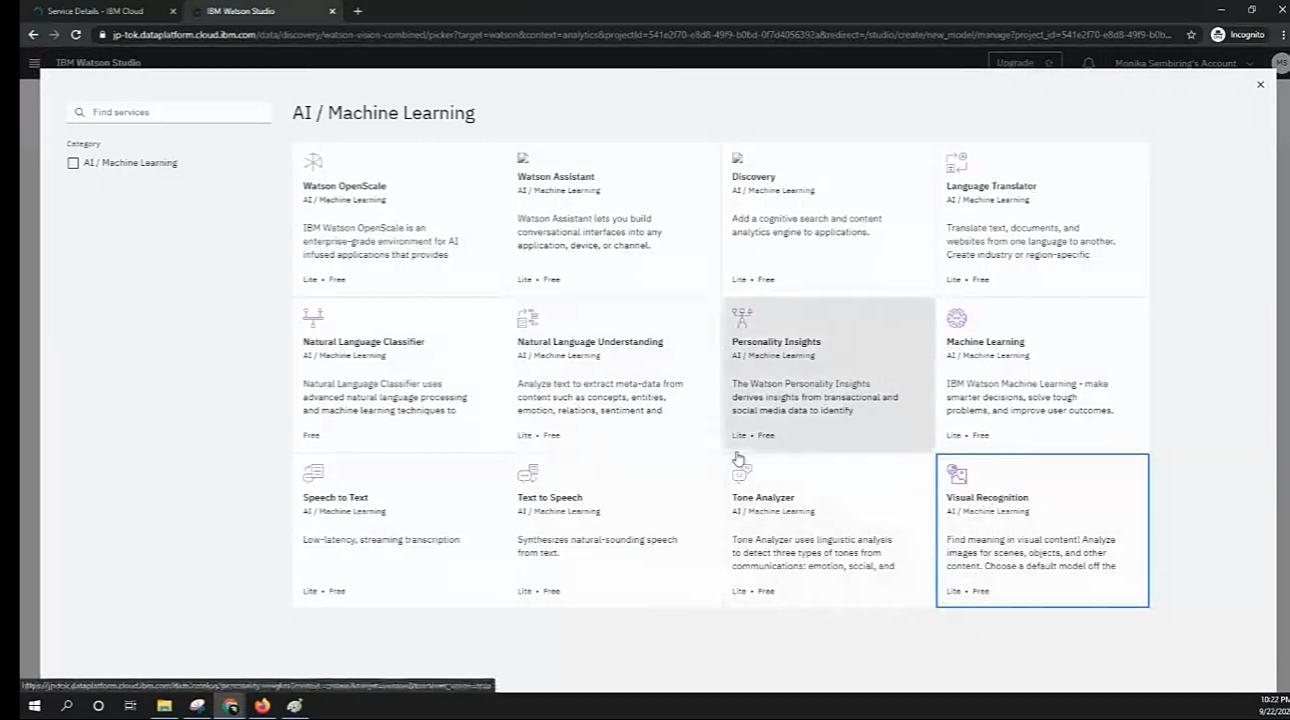
STEP 7 : CHOOSE ASSET TYPE as VISUAL RECOGNITION

STEP 8 : After choosen asset type then ASSOCIATE SERVICE OPEN

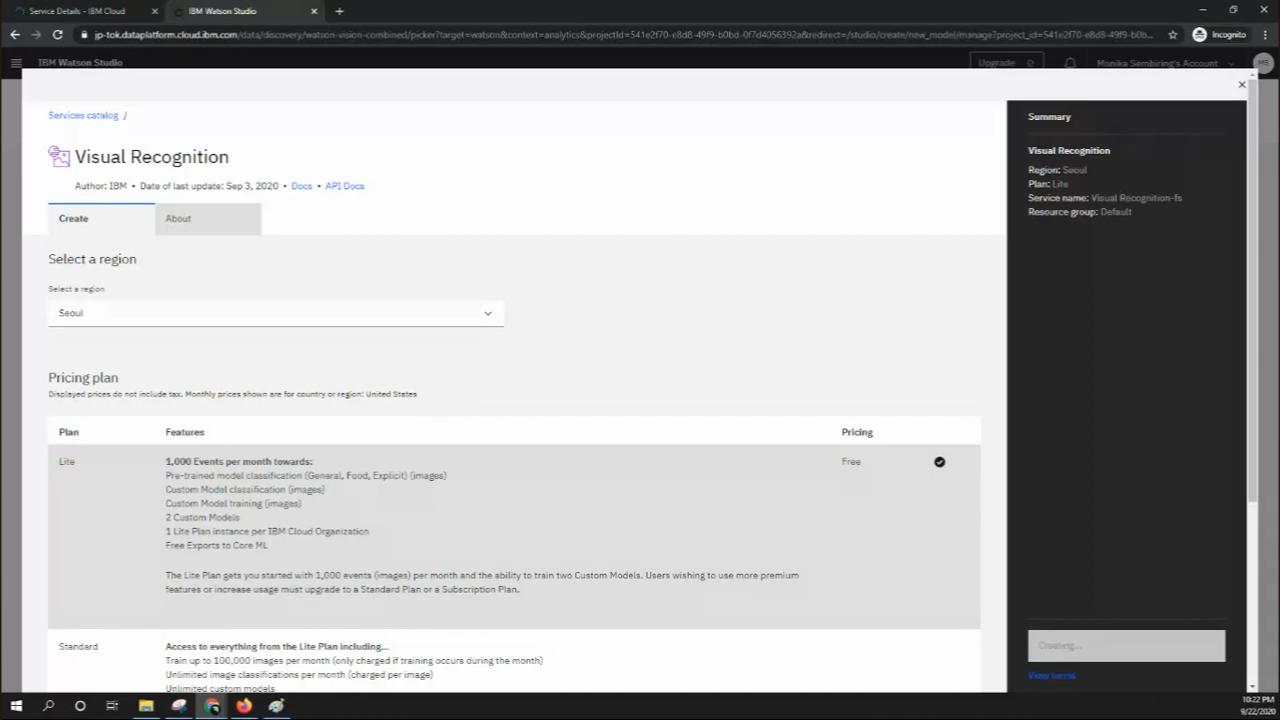
****

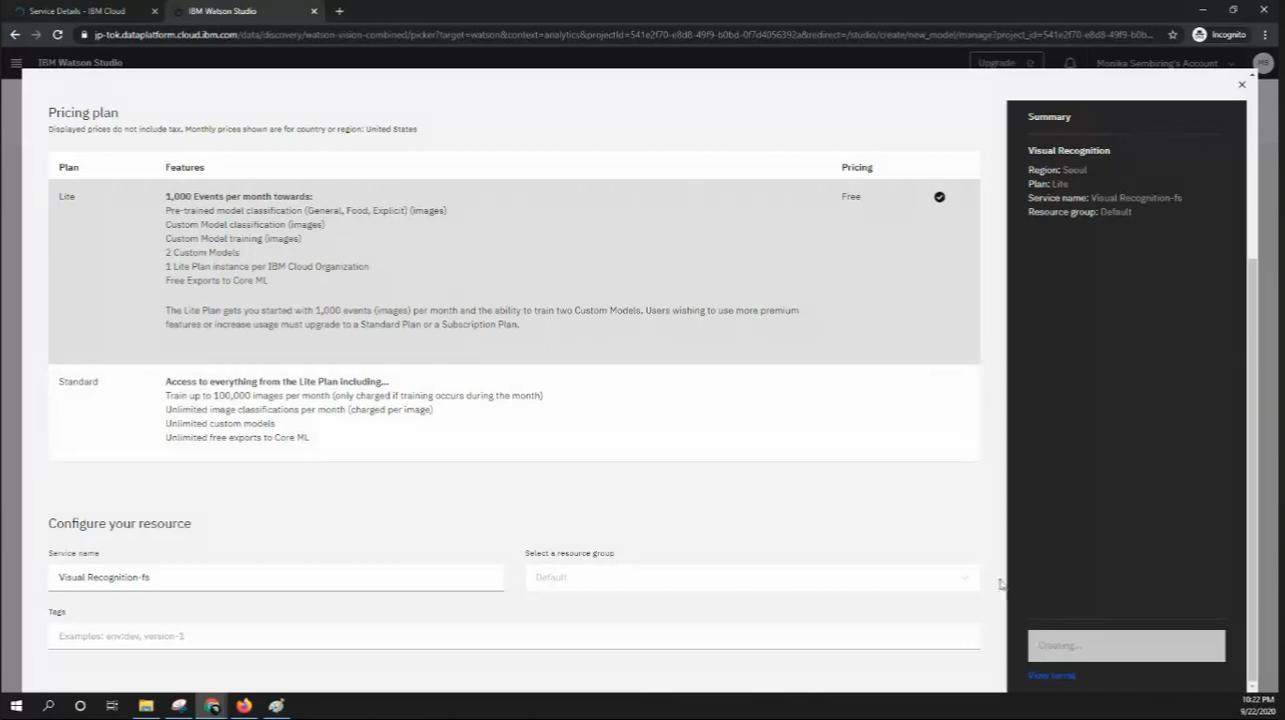
STEP 9 : IN ASSOCIATE SERVICE click NEW SERVICE and SELECT VISUAL RECOGNITION



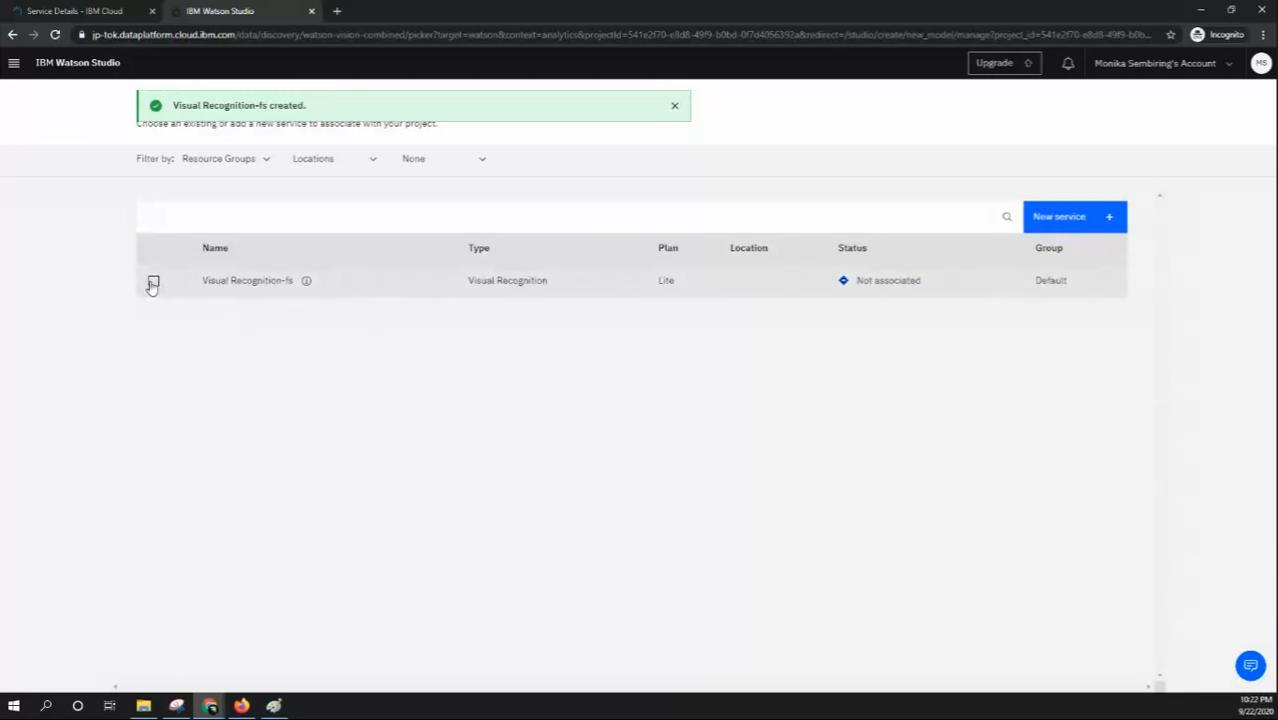


STEP 10 : AGAIN SET REGION AND CLICK CREATE

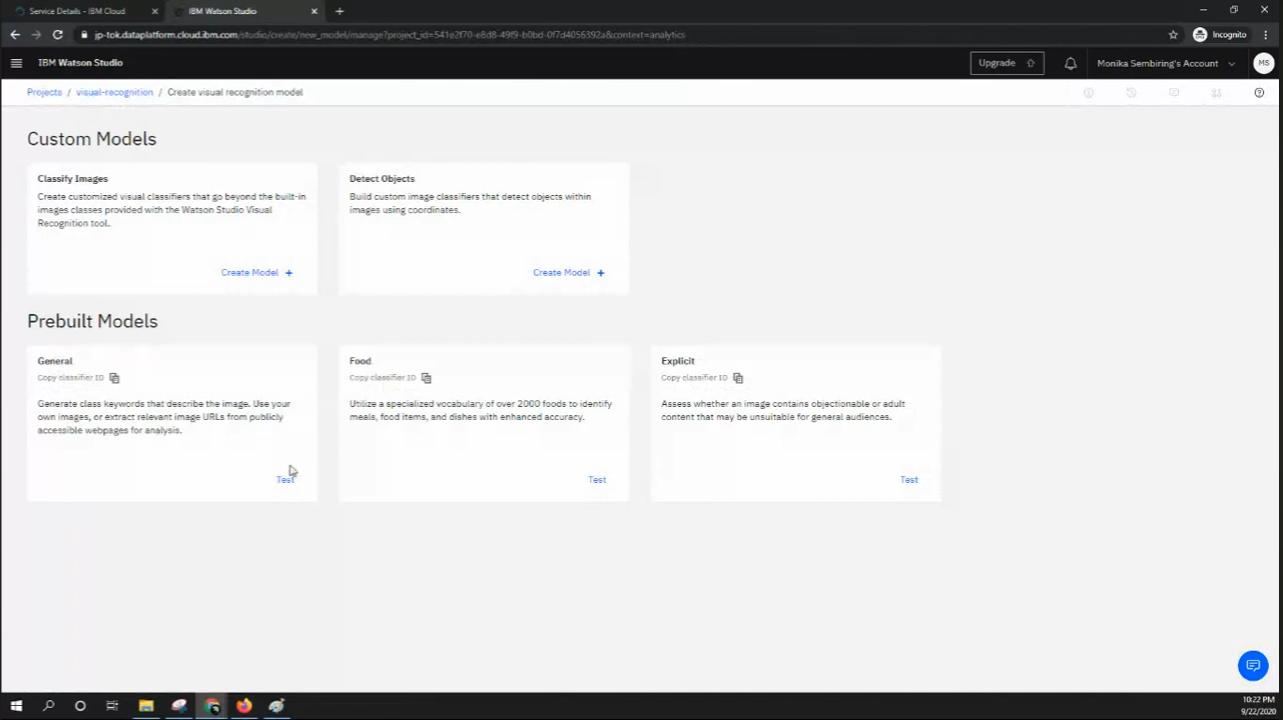




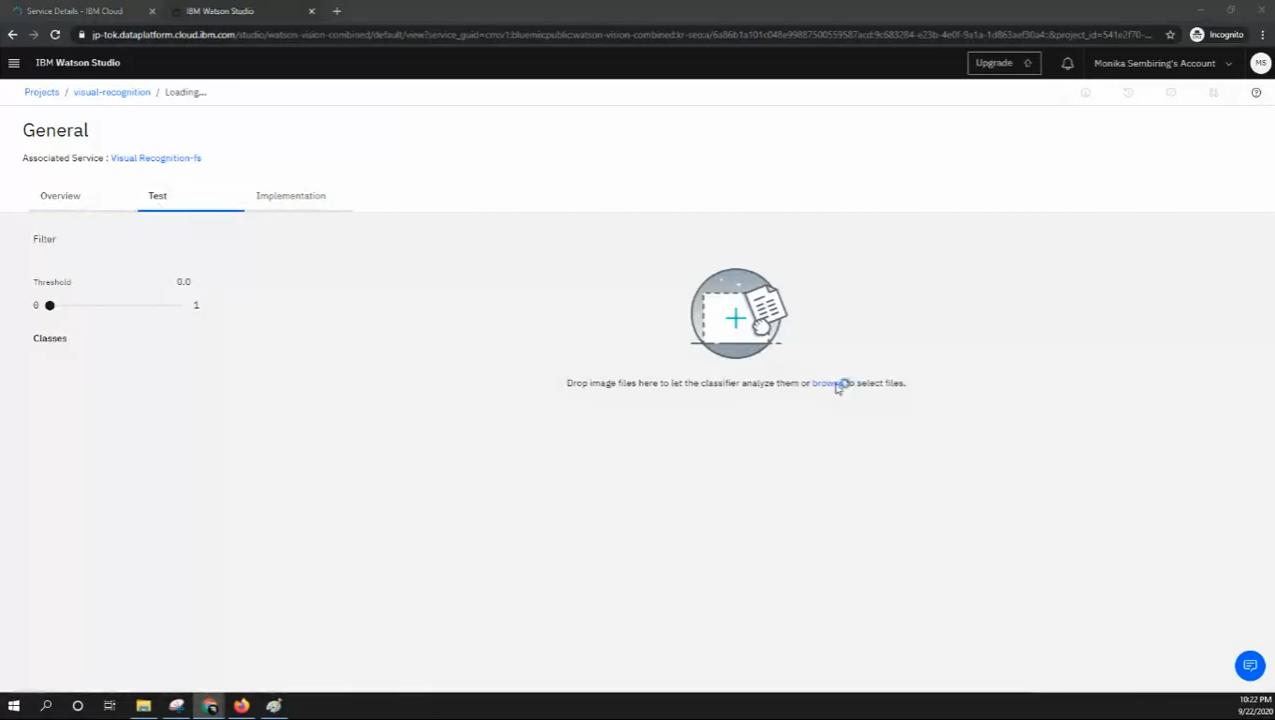
STEP 11 : SELECT VISUAL RECOGNITION and ASSOCIATE SERVICE

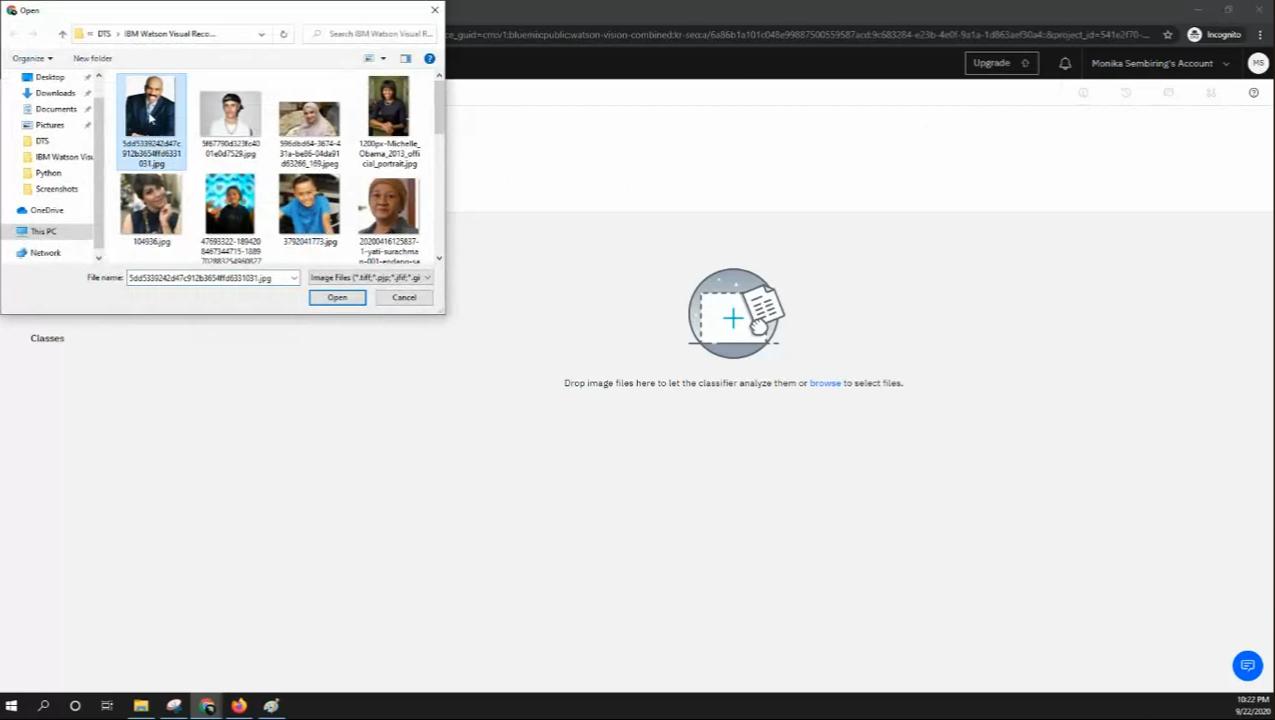


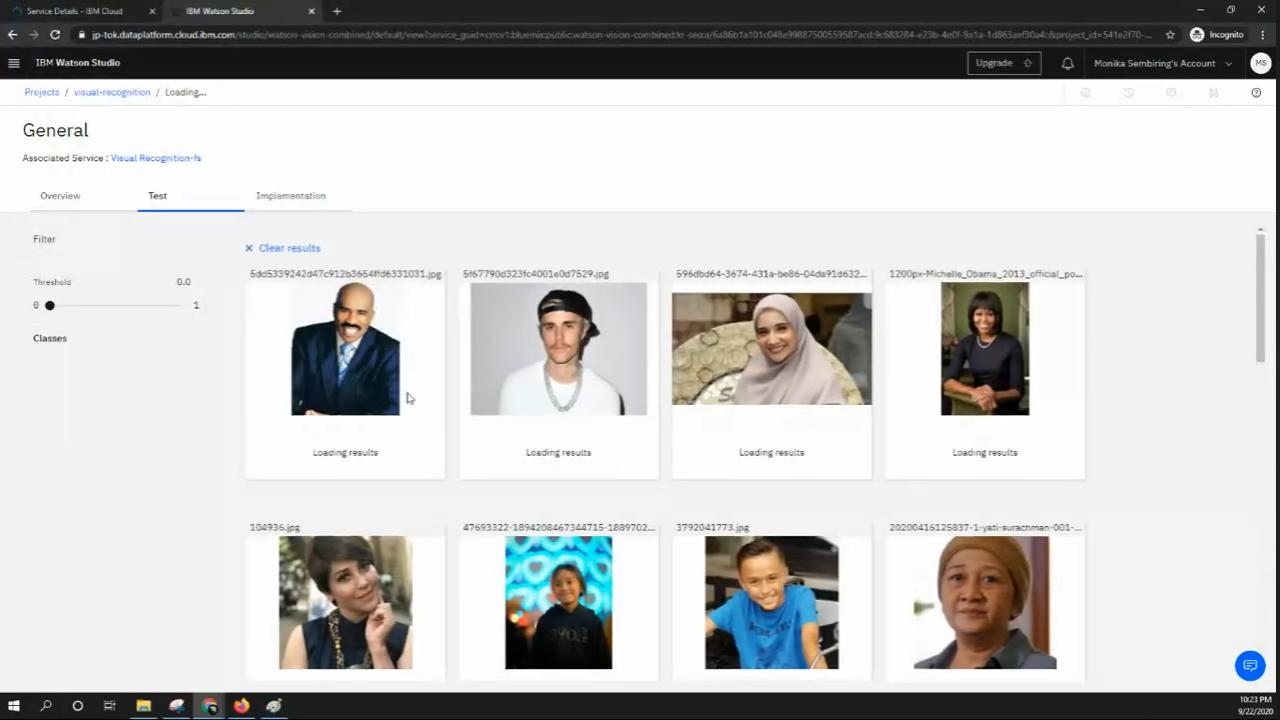
STEP 12 : CLICK TEST



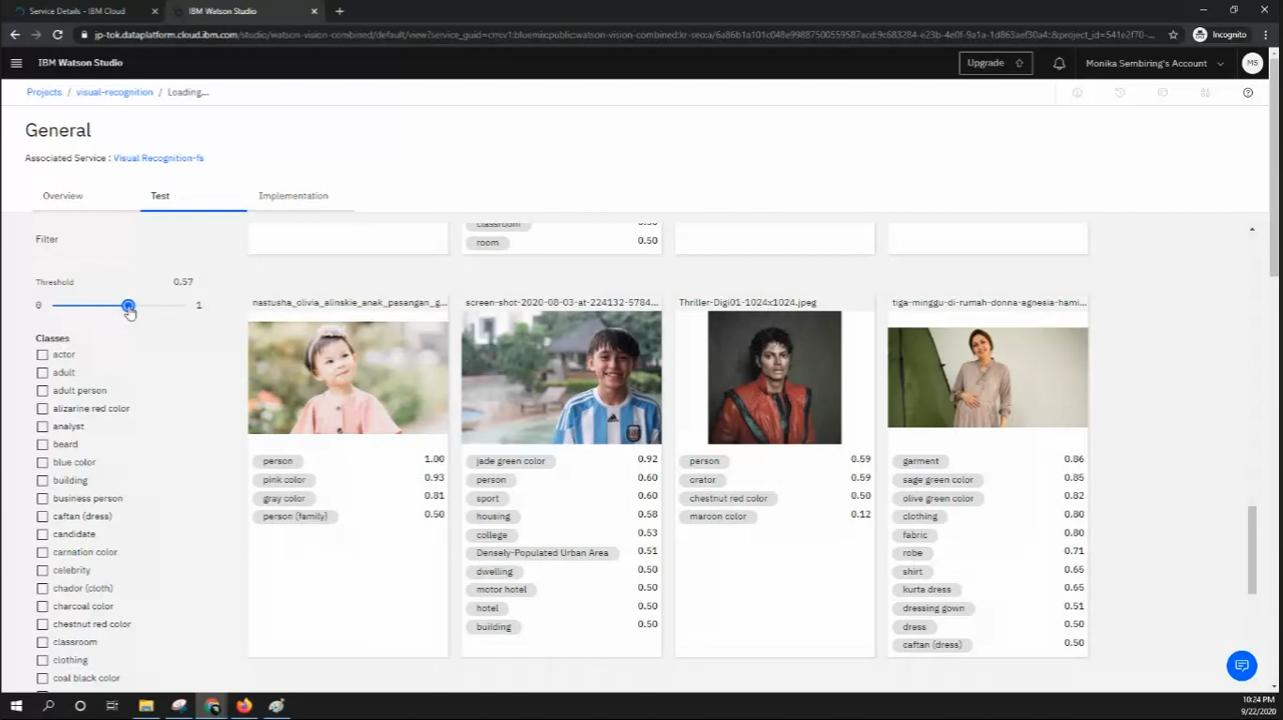
STEP 13 : SELECT FOLDER of Set Of Data Given to visual recognition







STEP 14 : We get the OUTPUT OF VISUAL RECOGNITION



# Use natural language generation to create captions for the recognized images.

Const form = document.querySelector(‘form’);

Const caption = document.querySelector(‘#caption’);

Form.addEventListener(‘submit’, async € => {

e.preventDefault();

const formData = new FormData(form);

const imageFile = formData.get(‘image’);

const response = await fetch(‘https://api.example.com/visual-recognition’, {

method: ‘POST’,

body: formData

});

Const data = await response.json();

Caption.textContent = data.caption;

});

FINAL CODING

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Image Recognition System</title>

  <style>

    /\* Apply some basic styles to the page \*/

    body {

      font-family: Arial, sans-serif;

      background-color: #f0f0f0;

      margin: 0;

      padding: 0;

    }

    header {

      background-color: #333;

      color: #fff;

      text-align: center;

      padding: 20px;

    }

    h1 {

      margin: 0;

    }

    section {

      text-align: center;

      margin: 20px;

    }

    form {

      display: inline-block;

    }

    input[type="file"] {

      display: none; /\* Hide the file input \*/

    }

    button {

      background-color: #3498db;

      color: #fff;

      padding: 10px 20px;

      border: none;

      cursor: pointer;

    }

    button:hover {

      background-color: #2980b9;

    }

    #result {

      background-color: #fff;

      padding: 20px;

      box-shadow: 0px 0px 5px 0px #888;

    }

    /\* Style the footer \*/

    footer {

      text-align: center;

      background-color: #333;

      color: #fff;

      padding: 10px;

      position: absolute;

      bottom: 0;

      width: 100%;

    }

    /\* Center the caption text \*/

    #caption {

      text-align: center;

    }

  </style>

</head>

<body>

  <header>

    <h1>Image Recognition System</h1>

  </header>

  <section>

    <form enctype="multipart/form-data" action="upload" method="post">

      <input type="file" name="image" accept="image/\*">

      <button type="submit">Upload</button>

    </form>

  </section>

  <section id="result">

    <h2>AI-Generated Caption:</h2>

    <p id="caption">Waiting for image recognition...</p>

  </section>

  <footer>

    <p>&copy; 2023 Image Recognition System. All rights reserved.</p>

  </footer>

  <script>

    const form = document.querySelector('form');

    const caption = document.querySelector('#caption');

    form.addEventListener('submit', async (e) => {

      e.preventDefault();

      const formData = new FormData(form);

      const imageFile = formData.get('image');

      const response = await fetch('https://api.example.com/visual-recognition', {

        method: 'POST',

        body: formData

      });

      const data = await response.json();

      caption.textContent = data.caption;

    });

  </script>

</body>

</html>

OUTPUT

