

WEEK 5 DEPLOYMENT OF MODEL AND API TO THE CLOUD

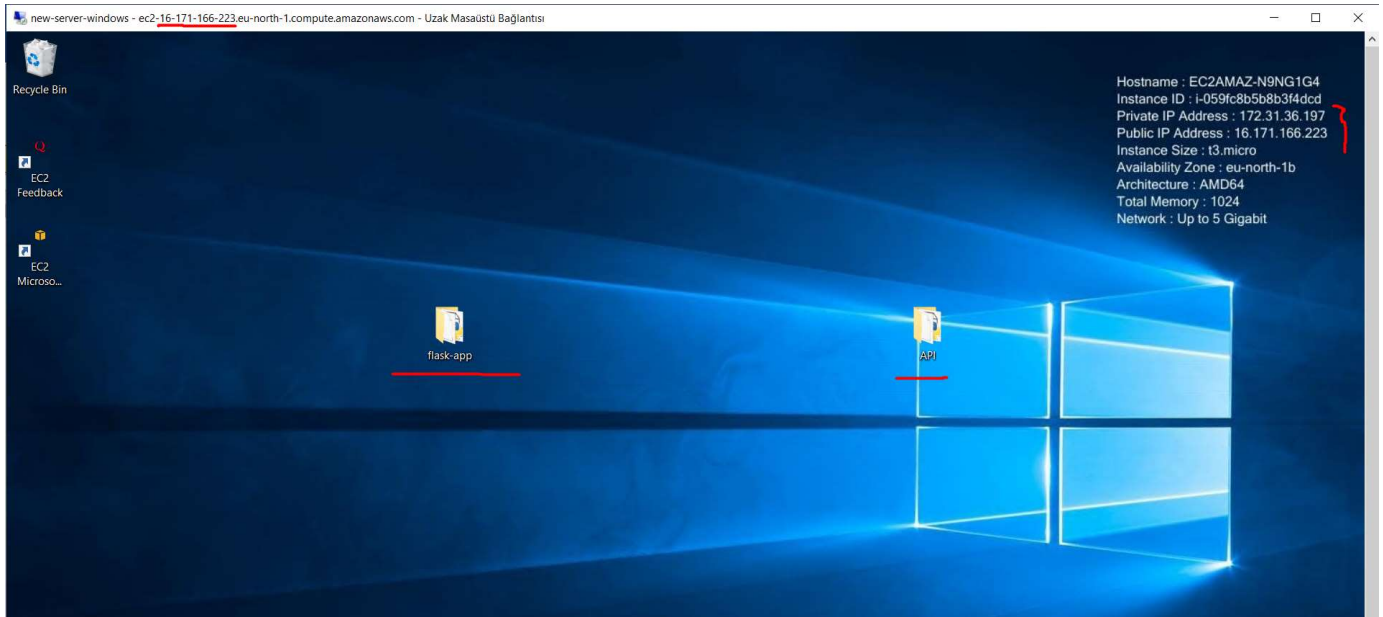
- A free account was created on AWS and a simple Windows AMI (Amazon Machine Image) was chosen for deployment.

The screenshot shows the AWS Management Console interface. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, and Snapshots. The main content area displays the 'Instances (1/1)' page. A table lists the instance 'new-server-wi...' with ID 'i-059fc8b5b8b3f4dcd', state 'Running', type 't3.micro', and '2/2 checks passed'. Below the table, the 'Instance: i-059fc8b5b8b3f4dcd (new-server-windows)' details are shown. The 'Instance summary' tab is active, displaying the instance ID, IP addresses, hostname type, and DNS names. The public IPv4 address is 16.171.166.223, and the public IPv4 DNS is ec2-16-171-166-223.eu-north-1.compute.amazonaws.com. The instance state is 'Running'.

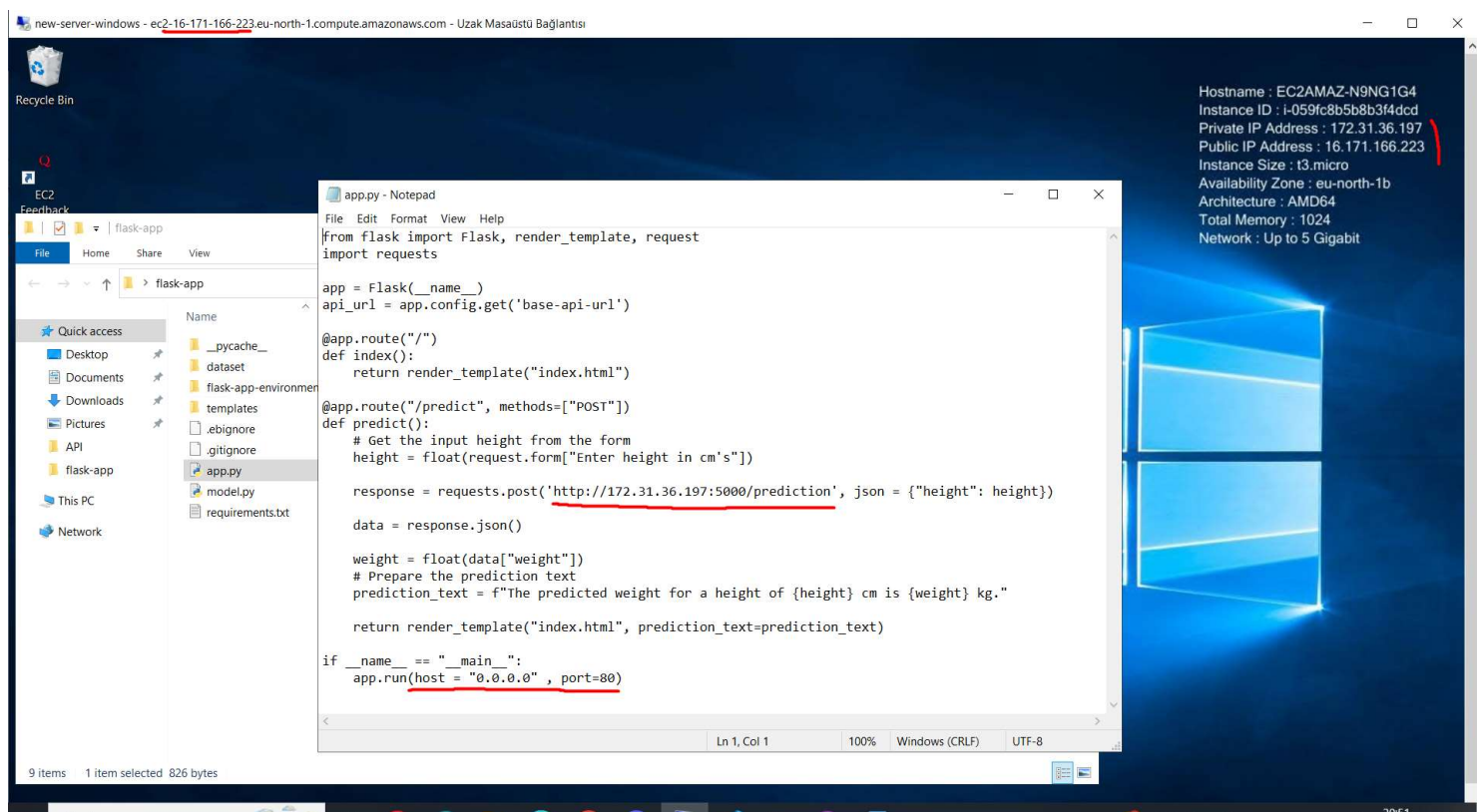
- Then, RDP (Remote Desktop File) Tool was downloaded for remote Access to the virtual machine (EC2 Instance).

The screenshot shows a Windows login window titled 'Windows Güvenliði'. It prompts the user to enter their identity information. The text indicates that the identity information will be used to connect to the EC2 instance with the public IP address 16.171.166.223. The user is prompted to enter a password for the Administrator user. Below the password field, the username 'DESKTOP-8N9CABT\Administrator' is displayed. There is a checkbox for 'Beni anımsa' (Remember me). At the bottom, there are two buttons: 'Tamam' (OK) and 'İptal' (Cancel). A link 'Daha fazla seçenek' (More options) is also visible.

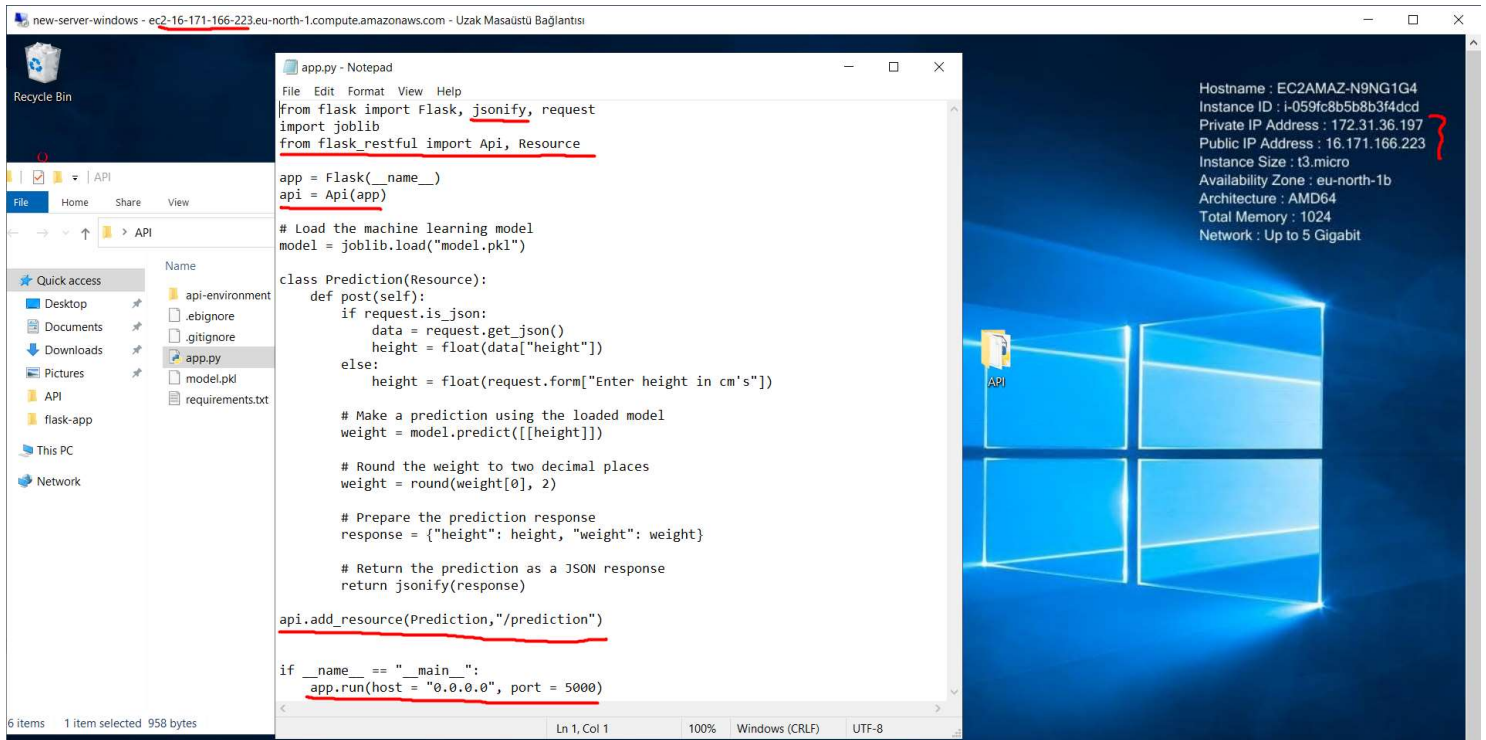
- After successful Access to the remote computer, 2 different folders “flask-app” and “API” was created to keep the , respectively, web app and the api.



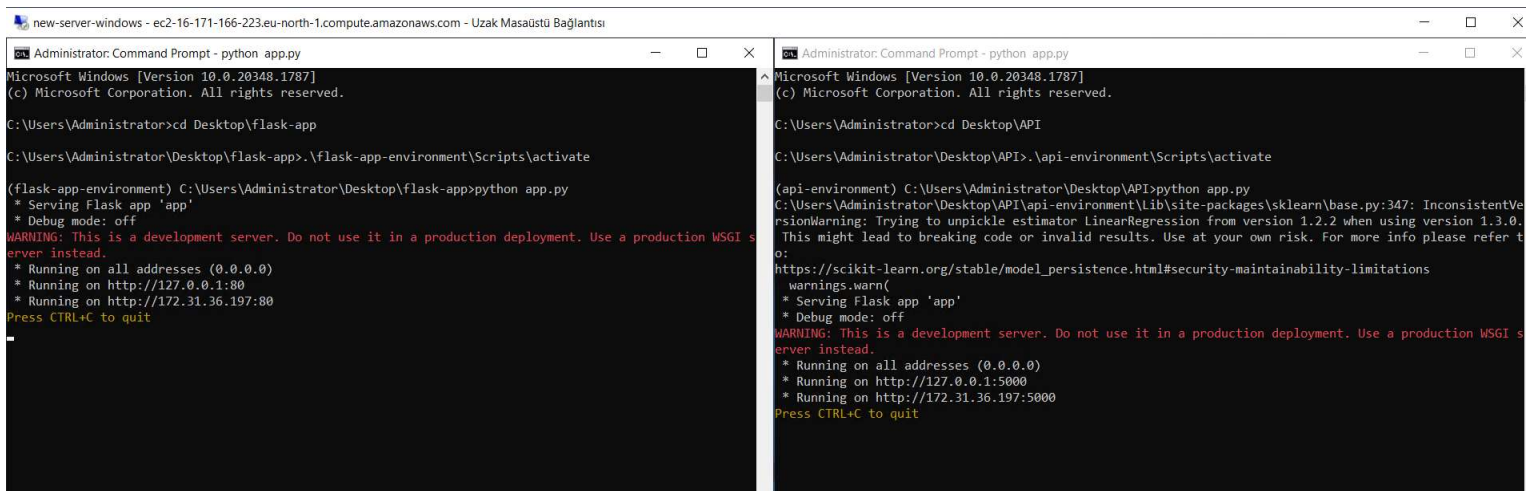
- After the creation of virtual environments, model prediction command and the model itself (model.pkl file) was seperated from this file and moved to the API file. Following changes for remote Access to the server was done (providing default way for host url and setting up the public IP for prediction of POST.)



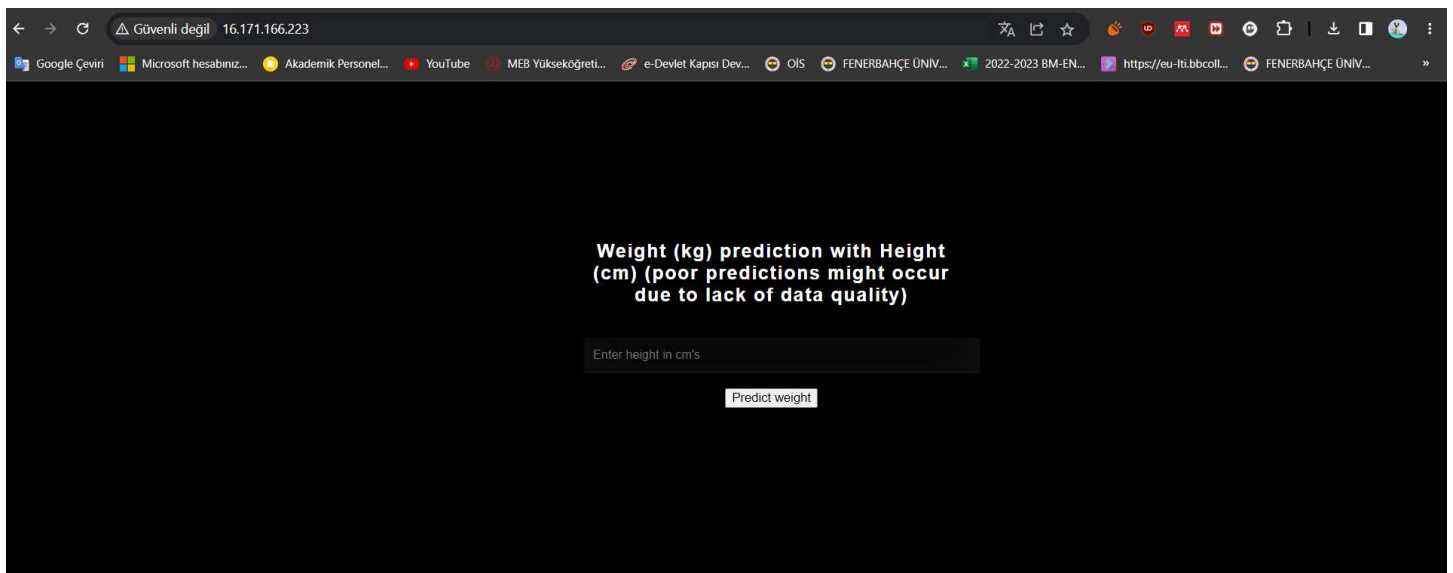
- In the API file, only the model and the code for making the prediction was held by setting up the API after downloading the Resource, API, flask_restful libraries:



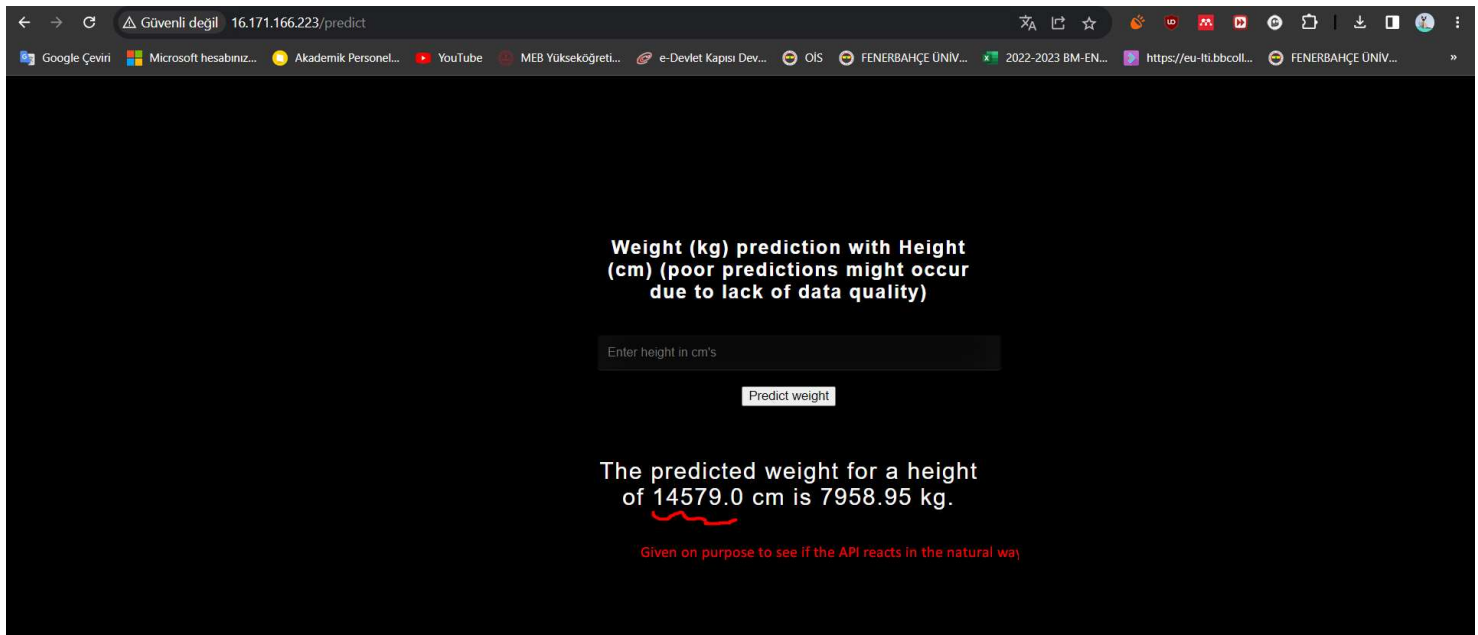
- Then, public IP Access was enabled from Windows firewall setting for third party users to be able to connect to it. Then, API and Flask App was run seperately as follow:



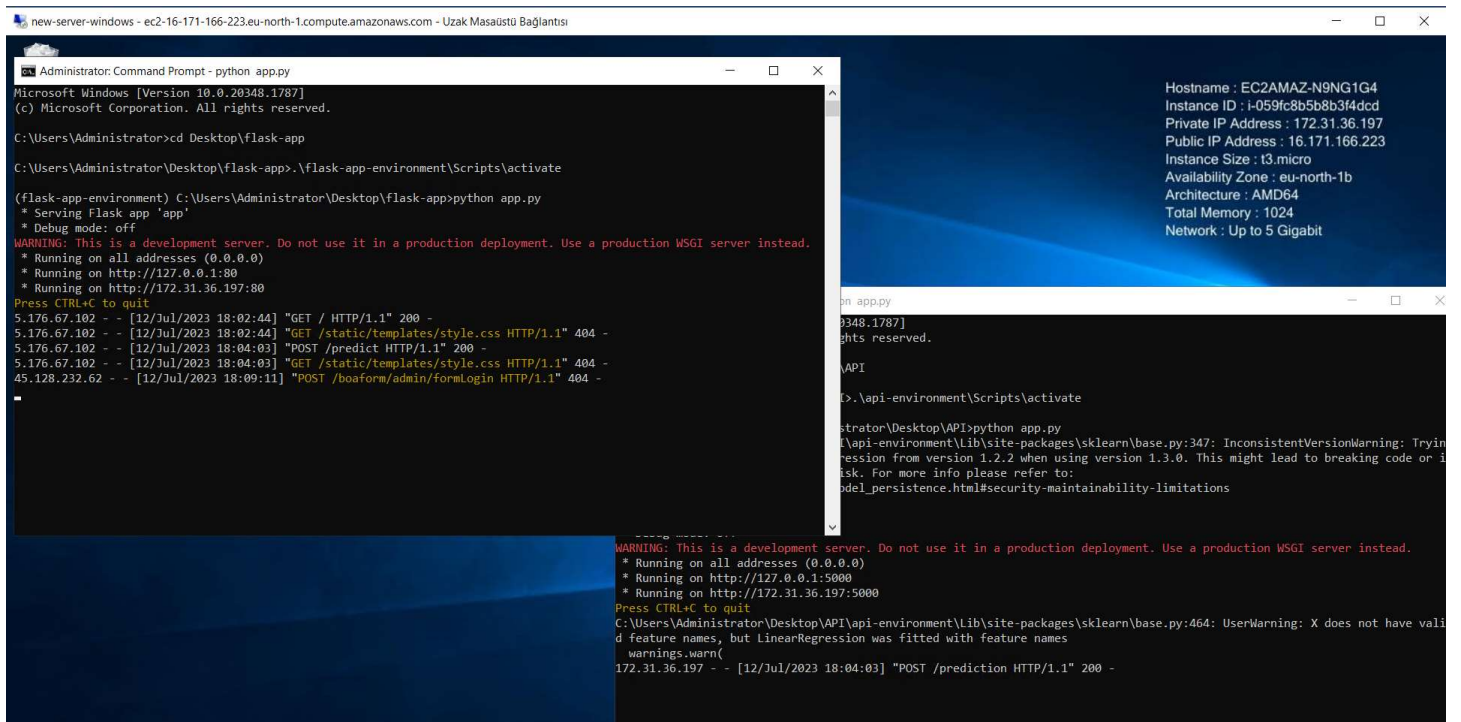
- Now, with a different computer, we are able to connect to public IP and Access the flask app and also the api itself for making predictions as follow:



- With predictions from remote server:



- Logs after a remote usage of the app.



Name : Batuhan YILMAZ
Batch Code : LISUM22
Submission Date : 12.07.2023
Submitted to : Github

DataGlacier Remote Internship on Data Science