Here's an example of how you can use the Azure Face API to perform face recognition:

1. First, you need to create a face group that will contain the known faces. You can use the "create" method of the Face - Large Person Group API to create a new face group. You need to specify a name and an ID for the group.
2. Once you have created the group, you can add known faces to it using the "add" method of the Face - Large Person Group Person API. You need to specify the group ID, a name and an ID for the person, and the image data (in binary format).
3. Once you have added the known faces to the group, you can use the "identify" method of the Face - Large Person Group - Identify API to identify a face in an image by comparing it to the faces in the group. You need to specify the group ID and the image data (in binary format).

The response from the API will contain an array of candidate faces, sorted by confidence. You can then parse the response to extract the information you need.

Please note that this is just an example and the actual implementation might require some adjustments based on your specific use case.

My code is a Python script that uses the Azure Face API to perform facial recognition tasks such as recognizing emotions and faces in an image, creating a face group, adding a known face to the group, and identifying a face in an image.

The script first imports the necessary libraries (requests and json) and defines a subscription key and endpoint for the Azure Face API. It then defines a function recognize\_emotion\_and\_face(image\_path) that takes an image path as an input and returns a dictionary of emotions and faces information or None if an error occurs. The function reads the binary data of the image, sends a POST request to the Azure Face API endpoint, and parses the JSON response.

After this, it creates a face group and adds a known face to the group using PUT and POST requests respectively. Then it uses POST request to identify a face in an image by passing largePersonGroupId as parameter, and the image data.

It then calls the function recognize\_emotion\_and\_face("Build\man.jpg") and prints the response and at last, it raises an exception if there's an error during any of the request made to Azure Face API.