1. Create a Database:

- Use PostgreSQL to make a database.

- Choose a plan and create a server.

- Make a new resource group.

- Assign a name to the server and pick a location.

- Choose the Python version.

- Add an admin username and set a password.

- Review your choices and click 'Create.' It will generate a server with details like admin name, storage, backup data.

2. Set Up Database Access:

- Go to your PostgreSQL database in your resources.

- Configure connections by adding an IP address range.

- Define the start and end IP addresses you want to allow access from.

- You can also include specific IP addresses if needed.

3. Connect Database with pgAdmin:

- Use a tool like pgAdmin to connect to your database server.

- Provide the server name as the host name and use the admin username.

- Create a database and save it. You can add schema and tables or write code.

4. Create a Web App:

- Make a web application to connect to your server.

- Name your app.

- Choose an app service plan.

- Define your Python code and deploy it.

5. Connect Web App to Database:

- In the web app settings, connect to the PostgreSQL database.

- Enter the host name, host port, password, admin username, and use FastAPI.

6. \*General Settings:

- Check and set your Python version in the general settings.and add startup code

7. Upload Code from GitHub:

- Upload your code from GitHub, including all necessary files.

- Set a starting command in general settings.

- Save your app server plan.

8. Deployment from GitHub:

- Open the deployment centre and select the GitHub option.

- Choose the file and repository for your app.

- Build and deploy your app from a specific branch.

9. Successful Deployment:

- After a successful deployment, you'll see a success message.

- Go to your web browser, access the Azure home, and run or execute your application. It will store data in your PostgreSQL database.