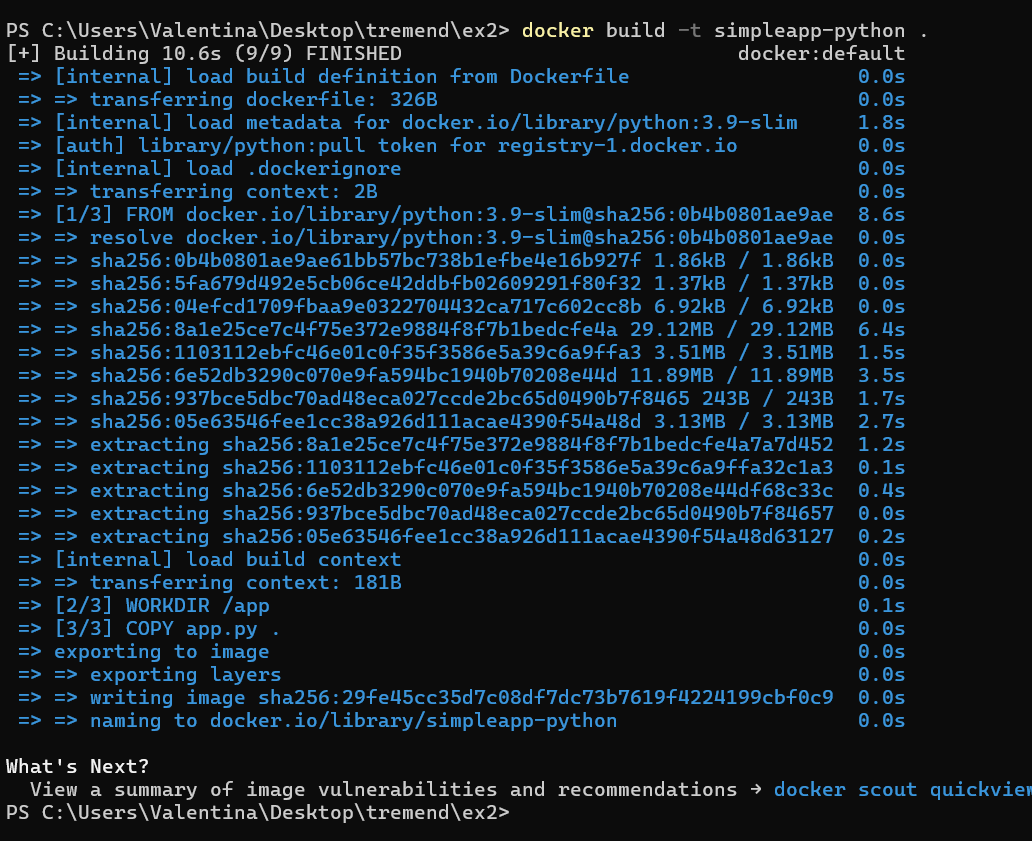
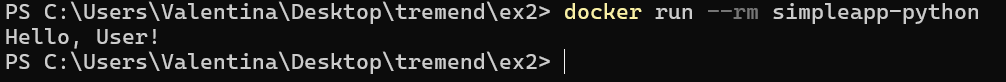
I began by selecting the Python application, app.py, which utilizes an environment variable to print a greeting message. Following this, I crafted a Dockerfile to encapsulate the Python application, ensuring the necessary dependencies were installed and the environment was properly configured.

Subsequently, I proceeded to build the Docker image locally using the following commands:

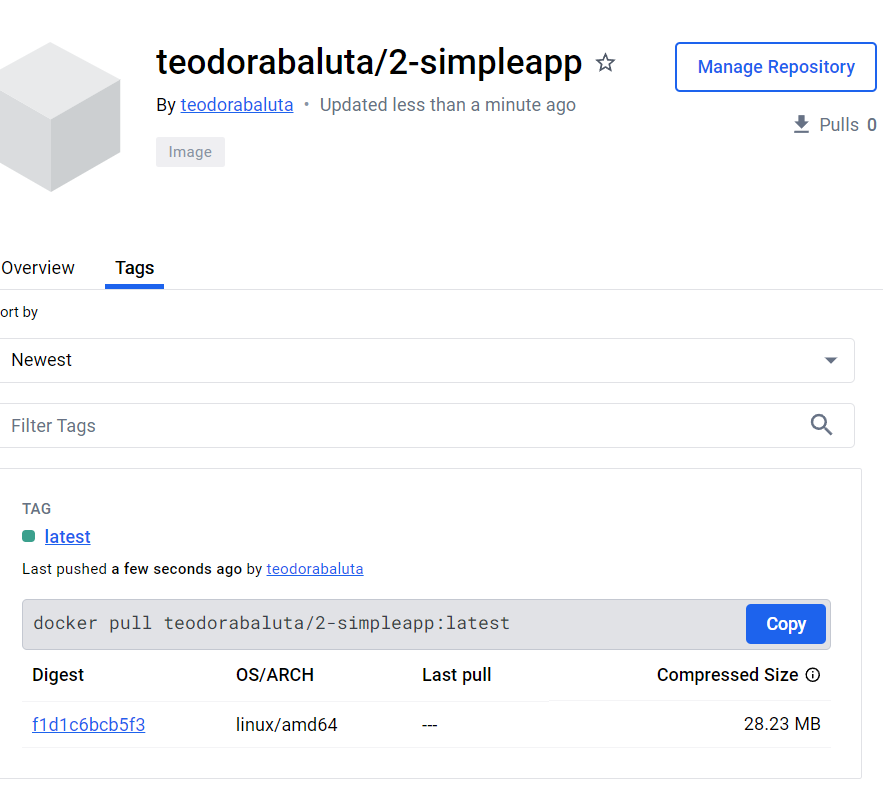




After logging into Docker with my credentials, I obtained the image successfully, as shown in the following screenshots.

A screen shot of a computer

Description automatically generated



Afterwards, within GitHub, I initiated the creation of a GitHub Actions workflow file named build-and-push-docker-image. This workflow was designed to automate the build and push processes associated with the Docker image. It encompassed steps to authenticate with Docker Hub, build the Docker image, tag it with version information, and subsequently push it to the Docker registry.

To facilitate secure authentication with Docker Hub, I securely stored Docker Hub credentials, including the username and password, as secrets within the repository settings of GitHub.

By following these steps, I successfully orchestrated the automation of building and pushing the Docker image, thereby streamlining the deployment process.

A screenshot of a computer

Description automatically generated