Yes, you can deploy a Docker application from one machine to another without uploading the Docker image to a public container registry like Docker Hub. This is often done using a process called "Docker image transfer" or "Docker image export/import." Here's how you can do it:

1. \*\*Export Docker Image:\*\*

On the source machine, export your Docker image into a tarball file:

```

docker save -o image.tar image-name:tag

```

2. \*\*Transfer the Image:\*\*

Transfer the exported `image.tar` file to the target machine using methods like SCP, SFTP, or any other file transfer mechanism you prefer.

3. \*\*Import Docker Image:\*\*

On the target machine, load the Docker image from the transferred tarball:

```

docker load -i image.tar

```

4. \*\*Run Container from Imported Image:\*\*

After loading the image, you can run a container from it on the target machine:

```

docker run -d -p host-port:container-port image-name:tag

```

This process allows you to move Docker images and containers between machines without using a public registry. However, keep in mind a few considerations:

- \*\*Image Compatibility:\*\* The source and target machines should have compatible architectures (e.g., both x86\_64) and operating systems. Docker images might not work properly if there are significant differences between the source and target environments.

- \*\*Dependencies:\*\* Make sure all the dependencies and libraries required by your application are available on the target machine. This is especially important if the machines have different environments.

- \*\*Security:\*\* This method skips the benefits of version control and distribution provided by container registries. Additionally, there are security concerns related to transferring images outside of a secure registry.

- \*\*Size and Speed:\*\* The size of the Docker image can impact the transfer time and the amount of disk space required on the target machine.

- \*\*Private Registries:\*\* If you have control over both machines, you could set up a private container registry on your network to facilitate image sharing in a more controlled manner.

While this method can be useful in specific scenarios, for a more robust and scalable deployment pipeline, it's generally recommended to use container registries (public or private) and deployment tools like Kubernetes, AWS ECS, or Docker Compose to manage and orchestrate your deployments.