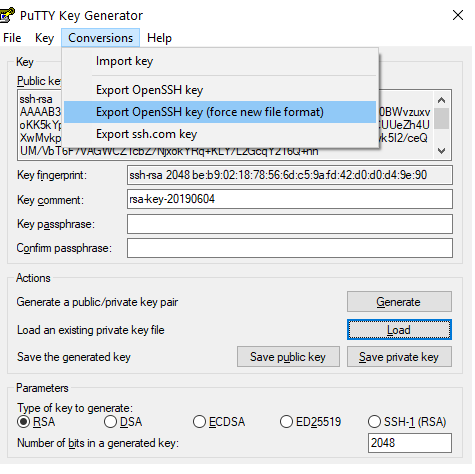
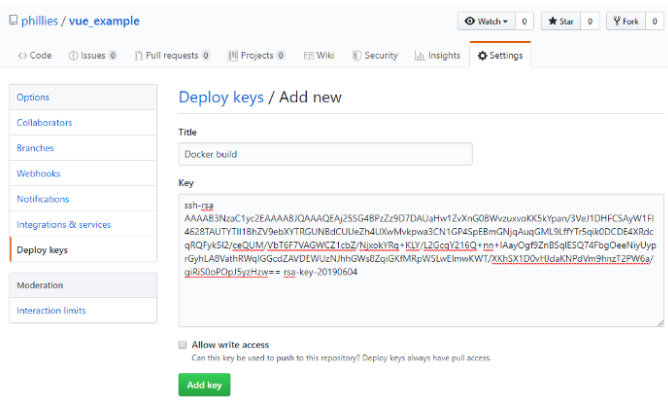
**Importing ssh keys into the builder to clone private repositories**

When you want to publish an image with an app from a private repository you need some way to pass credentials into the docker build process to allow git to log in to your repo. The direct and very dangerous approach would be to use username and password in the clone command. But you should never ever put credentials in a configuration file, so this is not an option. Passing the credentials as command line arguments into the builder is also not a good idea, as it requires to always pass the credentials and the password might be stored in bash history or some other log. The preferred solution are *deploy keys*. A set of ssh public-private keys that are specifically for read-only access to certain repositories. You can copy those keys into the builder, register them with the ssh agent, and then clone the repository. As the builder image is deleted after the process the final image does not contain any traces of the keys.

First you need to create an ssh key pair, a public key for github and a private key for you to use in the docker file. On linux just run ssh-keygen -t rsa -f github\_key to generate an RSA key pair. Make sure to not use a passphrase. Then copy the github\_key and github\_key.pub files next to your Dockerfile (for convenience). On Windows, PuTTY/[KiTTY](https://www.9bis.net/kitty/?page=Command-line%20options&zone=en" \t "_blank) can generate key pairs, make sure to export the private key in OpenSSH format.



Next you have to log in to github, navigate to your repository, select Settings, then Deploy keys, and then Add key.



There you can upload the content of your *public* key github\_key.pub and save the changes.

Now we just have to copy the github\_key into the builder image, start the ssh-agent (line 4), add the key to the agent (line 5), import the public key from github.com (line 6), and finally clone the image using ssh instead of https.



That’s it. Works with any kind of project and should also work with other git hosts as long as they support ssh logins.